PDE-Based Dynamic Data-Driven Monitoring of Atmospheric Dispersion Processes

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Vom Fachbereich Informatik der
Technischen Universität Darmstadt
zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte

Dissertation

von

Tobias Ritter, M.Sc.
(geboren in Hanau)

Referent:	Prof. Dr. Oskar von Stryk
Korreferent:	Prof. Dr. Stefan Ulbrich
Tag der Einreichung:	11.01.2017
Tag der mündlichen Prüfung:	28.03.2017

Darmstadt 2017
D17

Abstract

Atmospheric dispersion of pollutants highly affects human health and well-being. For disaster surveillance and response in such catastrophic scenarios, repeated estimation of the current process state and of relevant process parameters is essentially required. The respective estimates can be obtained by repeatedly integrating measurements from a sensor network into a process model capable of forecasting current estimates to future times. Networks of mobile sensors are increasingly considered in this context since they offer the possibility to move along trajectories, which provide a high expected information gain concerning the current dispersion state.

In related work, these trajectories are mostly determined offline prior to the application by solving a complex optimal control problem subject to process, vehicle and uncertainty dynamics. However, the trajectories depend on the process state and parameters to be determined. Hence, it is much more desirable to apply a dynamic data-driven approach, in which the available measurements are directly integrated into the process model and control inputs for the sensor-carrying vehicles are determined online based thereupon.

The dynamic data-driven application belongs to the domain of cyber-physical systems and is characterized by real-time requirements so that lightweight process models with rather inaccurate prediction abilities have been predominantly used in related work. On the other hand, models based on partial differential equations (PDEs) are typically related to the computationally expensive solution of a high-dimensional problem. However, due to their ability to provide physically realistic forecasts, PDE-models are a key feature of this thesis and simplifications are introduced to cope with them.

In this work, three new PDE-based dynamic data-driven monitoring strategies for state and parameter estimation of dispersion processes are proposed based on a coordinated fleet of sensor-carrying vehicles processing models and estimations online.

The first monitoring strategy is based on a sub-optimal procedural structure consisting of multiple loosely coupled building blocks to increase the efficiency compared to sophisticated optimal control problems. Forecasts stemming from a process simulation based on the finite element method are combined with the sensors' measurements using an efficient ensemble-based data assimilation method. A sequential procedure is proposed to identify informative measurement locations that are based on the current estimates as well as on their uncertainty. The identified locations are handed to a cooperative optimization-based vehicle controller, which is in charge of guiding the vehicles to the suggested locations. Furthermore, an estimation method to jointly estimate current process state vectors and the source function is presented and the handling of a model error covariance matrix depending on the process state is described.

In contrast to the first approach, the second approach is based on a decentralized computation and communication structure. In order to explicitly account for limited communication ranges and to increase scalability, a central computing node is waived and computations are performed only locally on-board the vehicles, exchanging local information with vehicles in their vicinity. While the general procedural structure of the centralized approach is preserved, modifications are necessary to account for the limited onboard computing capability, potentially different models maintained on different vehicles, and limited communication ranges. To simplify the process model, Proper Orthogonal Decomposition is used to generate reduced order models. A suitable choice of the snapshot set, required to generate the reduced order model, is proposed. Furthermore, a two-step reduced data assimilation method is developed updat-

ing the own local state with the local measurement first and fusing state information with neighbors in a second step. A decentralized version of the identification procedure for suitable measurement locations is described and an existing decentralized variant of the vehicle controller is employed.

The third approach represents a partitioned decentralized dynamic data-driven monitoring strategy. To further pursue decentrality, the global problem domain is decomposed into several subdomains with vehicles being assigned to subdomains they are currently located in. In this way, only a local process and multi-vehicle model has to be maintained by each sensor-carrying vehicle, while communication with vehicles on directly adjacent subdomains ensures convergence of the whole problem. A new ensemble-based prediction and update method is proposed that is based on an efficient domain decomposition method and that requires only minimal exchange between vehicles on neighboring subdomains. For more flexibility, target point generation and vehicle trajectories are adapted to allow vehicles to move to other subdomains, where they might be even more useful.

The respective approaches are applied in a number of test case scenarios and are evaluated regarding their estimation results and computing time. While the centralized monitoring approach provides the best estimation results, significantly outperforming other basic monitoring strategies, limited communication ranges, a large number of mobile sensors and higher dimensional problems could become problematic. In the latter case, the decentralized approach offers a more efficient alternative at the cost of minor losses in estimation quality. The same applies to the application of the partitioned approach, which, while being less flexible, provides even more scalability with respect to the number of sensor-carrying vehicles.

Zusammenfassung

Die Ausbreitung von Schadstoffen in der Atmosphäre kann Gesundheit und Wohlbefinden stark beeinträchtigen. In Katastrophenszenarien wird daher zum Zwecke der Überwachung und des Katastrophenschutzes eine Schätzung des aktuellen Zustandes und relevanter Parameter des Ausbreitungsprozesses essenziell benötigt. Die jeweiligen Schätzungen können dabei durch eine fortwährende Integration von Messdaten eines Sensornetzwerks in ein Vorhersagemodel des Prozesses gewonnen werden. Zunehmend werden in diesem Kontext Netzwerke aus unbemannten Fahrzeugen betrachtet, die mit geeigneten Sensoren ausgestattet sind. Diese bieten die Möglichkeit, entlang von Trajektorien zu messen, die bezogen auf die aktuelle atmosphärische Ausbreitung einen besonders hohen erwarteten Informationsgewinn liefern.

In bisherigen Ansätzen werden solche Trajektorien meist offline vor der Anwendung durch Lösen eines komplexen Optimalsteuerungsproblems mit Prozess-, Fahrzeug- und Unsicherheitsdynamik in den Nebenbedingungen bestimmt. Allerdings hängen diese Trajektorien von dem zu bestimmenden Prozesszustand bzw. den zu bestimmenden Prozessparametern ab. Es ist daher erstrebenswert einen dynamisch datengetriebenen Ansatz anzuwenden, in dem die verfügbaren Messungen direkt in das Prozessmodell integriert werden und die Steuereingänge der mit Sensoren ausgestatteten Fahrzeuge online darauf basierend bestimmt werden.

Dynamisch datengetriebene Anwendungen gehören zur Gruppe der cyber-physischen Systeme, die sich durch Echtzeitanforderungen auszeichnen. Aus diesem Grund wurden bisher hauptsächlich stark vereinfachte Prozessmodelle mit niedrigem Rechenzeitbedarf, aber eher ungenauen Vorhersagequalitäten benutzt. Modelle, die auf partiellen Differentialgleichungen (PDEs) beruhen, benötigen zwar die rechenintensive Lösung eines hochdimensionalen Problems, besitzen allerdings auch die Eigenschaft physikalisch plausible Vorhersagen treffen zu können. Um die Vorteile solcher PDE-Modelle für die betrachteten Anwendungsszenarien nutzen zu können, sind daher eine Reihe von Annahmen und Schritten nötig, die in dieser Arbeit beschrieben werden.

Insgesamt werden in dieser Arbeit drei neue PDE-basierte dynamisch datengetriebene Monitoringstrategien zur Zustands- und Parameterschätzung atmosphärischer Ausbreitungsprozesse entwickelt, jeweils basierend auf einer Gruppe von kooperativ arbeitenden und mit Sensoren ausgerüsteten Fahrzeugen, die Modelle und Schätzungen online verarbeiten.

Die erste Strategie besteht aus mehreren lose gekoppelten Schritten, um die Berechnungseffizienz im Vergleich zur Lösung eines komplexen Optimalsteuerungsproblems zu steigern. Vorhersagen, die auf einer Prozesssimulation mit der Finiten Elemente Methode beruhen, werden mit den Messungen der Sensoren unter Benutzung einer effizienten, ensemblebasierten Datenassimilationsmethode kombiniert. Um informative Messpositionen auf Basis der aktuellen Schätzungen und deren Unsicherheit zu identifizieren, wird ein sequentieller Algorithmus vorgeschlagen. Die so identifizierten Positionen werden an einen kooperativen, optimierungsbasierten Fahrzeugregler übergeben, der die Aufgabe hat, die Fahrzeuge zu den vorgeschlagenen Messpositionen zu führen. Außerdem wird eine Methode zur gemeinsamen Schätzung des aktuellen Prozesszustandes und der Quellfunktion sowie die Behandlung einer zustandsabhängigen Modellfehlerkovarianzmatrix eingeführt.

Der zweite Ansatz unterscheidet sich durch eine dezentrale Berechnungs- und Kommunikationsstruktur. Um eine begrenzte Kommunikationsreichweite einzubeziehen und die Skalierbarkeit zu erhöhen,

wird dabei auf einen zentralen Rechenknoten verzichtet und Berechnungen werden nur lokal auf den Fahrzeugen durchgeführt, wobei nur lokale Informationen zwischen benachbarten Fahrzeugen ausgetauscht werden. Während die allgemeine Ablaufstruktur des zentralen Ansatzes erhalten bleibt, sind Änderungen in den Teilschritten zur Berücksichtigung der begrenzten Rechenleistung an Bord der Fahrzeuge, der potentiell voneinander verschiedenen Modelle an verschiedenen Fahrzeugen und des limitierten Kommunikationsradius nötig. Zur Vereinfachung des Prozessmodells wird die sogenannte Proper Orthogonal Decomposition eingesetzt, um reduzierte Modelle zu generieren. Eine passende Wahl der Offline-Simulationen, die benötigt werden, um das reduzierte Modell zu erstellen, wird vorgeschlagen. Weiterhin wird eine auf diesen reduzierten Modellen beruhende, zweistufige Datenassimilationsmethode entwickelt, die zunächst die lokale Schätzung mit der lokalen Beobachtung kombiniert, um anschließend die entstandene Schätzung mit der Schätzung der Nachbarfahrzeuge zu fusionieren. Eine dezentrale Version der Zielpunktgenerierung wird beschrieben und eine bereits existierende dezentrale Variante des kooperativen Fahrzeugreglers wird angewendet.

Der dritte Ansatz stellt eine partitionierte dezentrale dynamisch datengetriebene Monitoringstrategie dar. Dabei wird das globale Problemgebiet in mehrere Teilgebiete aufgeteilt und die Fahrzeuge werden den Teilgebieten zugeordnet, in denen sie sich gerade befinden. Auf diese Weise muss nur ein lokales Prozess- und Sensormodel an jedem Knoten benutzt und gespeichert werden, während Kommunikation mit Fahrzeugen auf direkt benachbarten Teilgebieten die Konvergenz der Gesamtlösung gewährleistet. Eine neue ensemblebasierte Vorhersage- und Update-Methode wird vorgeschlagen. Diese Methode beruht auf einer effizienten Gebietszerlegungsmethode, die nur minimalen Austausch zwischen Fahrzeugen auf benachbarten Gebieten benötigt. Zur Erhöhung der Flexibilität werden Zielpunktgenerierung und Fahrzeugregelung so angepasst, dass es Fahrzeugen möglich ist in andere Teilgebiete zu fahren, in denen ihre Messungen deutlich dringender benötigt werden.

Die jeweiligen Ansätze werden in einer Reihe von verschiedenen Testfallszenarien angewendet und hinsichtlich ihrer Schätzergebnisse und der benötigten Rechenzeit evaluiert. Während der zentrale Monitoringansatz die besten Schätzergebnisse liefert und dabei andere grundlegende Monitoringstrategien aussticht, können die Nichtberücksichtigung von begrenzten Kommunikationsreichweiten, eine große Anzahl von Sensoren und höherdimensionale Zustandsvektoren zu Problemen führen. In den letzteren Fällen ist daher der dezentrale Ansatz auf Kosten von geringen Einbußen in der Schätzqualität eine effizientere Alternative. Dasselbe gilt für die Anwendung des partitionierten Ansatzes, der zwar weniger flexibel ist, bezüglich der Anzahl der Fahrzeuge und der Größe des Problemgebiets aber ein Plus an Skalierbarkeit mit sich bringt.

Acknowledgements

I would like to thank the many people who – directly and indirectly – have helped and supported me during the development of my thesis.

First of all, I wish to express my gratitude to my first supervisor Prof. Dr. rer. nat. Oskar von Stryk for giving me the opportunity to work in this interesting field of research and for the support over the years. He helped me to shape my work giving me a lot of freedom for my own ideas and approaches, and made it possible to pursue and finalize my thesis during my time as research assistant at the Simulation, Systems Optimization and Robotics Group (SIM).

Likewise, I would like to thank Prof. Dr. rer. nat. Stefan Ulbrich not only for being the second referee of my thesis but also for the constant technical input. He always supplied me with new ideas and provided solutions when I got stuck.

I gratefully acknowledge the support of the 'Excellence Initiative' of the German Federal and State Governments and the Graduate School of Computational Engineering (GSC CE) at TU Darmstadt. In this context, I would like to thank the staff of the GSC CE for providing me an excellent environment for my studies.

Likewise, I would like to thank the staff at SIM for their administrative and organizational help.

Moreover, thanks to all of my current and former colleagues at GSC CE and SIM, who helped and supported me in many different ways. Thanks for introducing me to new interesting topics, for discussing and giving me feedback concerning various aspects of my work, for proofreading my thesis, for checking my final presentation, and, most importantly, for the good atmosphere and the good time, which I will never forget. In particular, I would like to thank Juliane Euler for the fruitful technical cooperation and for sharing doubts and motivation during the final phase.

Most of all, I wish to thank my family. Thanks for the constant and comprehensive support – during my doctoral studies and during my whole life.

Contents

1 Introduction

1.1 Atmospheric Dispersion of Pollutants

Atmospheric dispersion of pollutants represent a key threat for human life and well-being. Pollutants, hazardous material, and other harmful chemical substances are released after accidents or natural disasters, disperse in the air and affect human health, life, and environment – partially far away from the actual disaster site.

An example for hazardous atmospheric dispersion processes are the incidents in Fukushima. In March 2011, an earthquake caused a meltdown in the Daiichi nuclear reactor. Nuclear substances were released in the air and were transported regionally and globally by the wind (see Figure 1.1(a)). As a consequence, several ten thousands of people had to be evacuated and radioactive deposition contaminated groundwater and agricultural products [169].

Fukushima was, of course, no solitary case. Comparable effects took place after the nuclear incidents in Chernobyl and Kyshtym in the 20th centuries where the atmospheric dispersion of nuclear substances claimed thousands of victims [166].

Another dispersion example is the eruption of the Icelandic volcano Eyjafjallajokull in 2010 (see Figure 1.1(b)). A gigantic ash plume led to the collapse of the European air travel for several days and affected economy and human well-being [72].

(a) Plume caused by the Fukushima incidents, extracted from [176]

(b) Volcanic ash plume caused by the eruption of Eyjafjallajokull, extracted from [157]

Figure 1.1: Exemplary atmospheric dispersion processes

Besides these major incidents on global scale, there are many incidents with impacts on smaller scales with typical radii from a few meters to several kilometers. Frequently, an accidental explosion or a leakage of a gas pipe of a chemical plant is the reason for hazardous substances being released in the air and threatening population living nearby. Examples are the Bhopal disaster in 1984, where highly toxic

substances leaked from a pesticide plant leading to the death of thousands of people, and the explosions at the Port of Tianjin in 2015.

Even in Germany, a number of different examples can be found. In 2012, a mass evacuation was caused by a poisonous cloud emitted from a leak at the Kraft Foods plant in northern Germany and, very recently, slight gas leakages occurred after the explosions at the BASF plant in Ludwigshafen.

The OECD predicts that the number of disasters will even increase in future [130]. Although technology is continuously improving and some risks are reduced, new technology with new risks, increasing population, population concentration in metropolitan areas and changing environmental conditions are likely to lead to more accidents. Furthermore, the current political situation leads to a higher risk of incidents caused by terrorist attacks, with power plants and chemical factories providing attractive targets. Monitoring of atmospheric dispersion processes that enables the assessment of impacts is, thus, a challenging and highly relevant task to be faced in the 21st century.

1.2 Dynamic Data-Driven Monitoring of Dispersion Processes

Societies have to be prepared for threads related to atmospheric dispersion and strategies have to be developed to support disaster response. In particular, it is of paramount interest for disaster response to monitor the dispersion process to be able to assess possible future impacts of the dispersion. Monitoring of atmospheric dispersion processes is mainly aimed at answering the following questions:

- *State estimation:* How large is the current global pollutant concentration?
- *Targeted state estimation:* How large is the current concentration at specific locations of interest?
- *Parameter identification:* What are the current values of important process parameters (e.g. diffusivity, reaction terms, etc.) and how do they evolve over time?
- *Source identification:* Where is the source located and how large is its output?
- *Prediction:* How does the process evolve in the future?

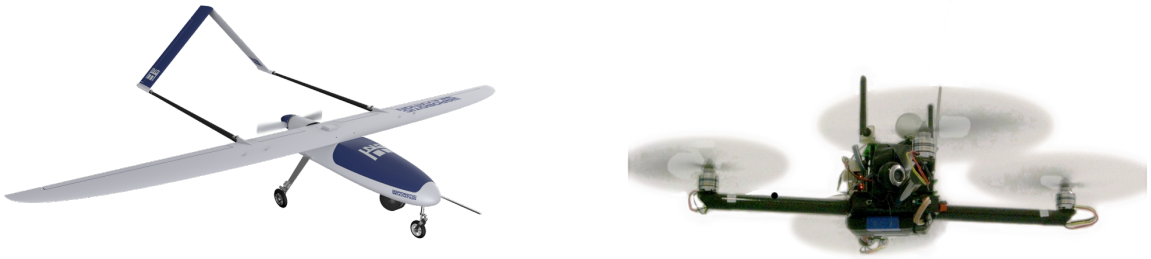
Having estimates of important process entities, the problem can be analyzed on a well-founded basis and the right steps to respond to the catastrophe can be initiated.

Basis of the monitoring is a model of the physics of the considered dispersion process. In this way, important entities can be identified and coherences of the considered processes can be described. Besides stochastic and heuristic models, *partial differential equation* (PDE) models are frequently used in this context. The latter allow precise, physically based predictions since they consider the evolution of the physics of the current process.

However, a model alone is not sufficient for a monitoring result with good accuracy. Uncertainties in model parameters, initial conditions, boundary conditions as well as errors stemming from external perturbations that are not modeled ask for another source of information. Therefore, sensor networks are applied in context of atmospheric dispersion monitoring. The sensors are deployed over the domain of interest and provide measurements of interesting process entities. Sensor data and model information can be repeatedly combined and a current estimate of the interesting process entities is available at every time.

An increasing trend is the use of robots for environmental monitoring [183, 53]. This means that *Unmanned Aerial Vehicles* (UAVs), for example quadrocopters or fixed-wing aircrafts (see Figure 1.2), are equipped with sensors and computing devices to provide measurements for the monitoring process. Vehicles being in charge of this task are referred to as *sensor vehicles* in the remainder of this work.

Compared to purely static sensor networks, mobile sensor networks are much more flexible, can cover a much larger domain and can move along trajectories that provide a particularly high information gain for measurements. The latter leads to a concept called *Adaptive Observation* (AO), describing the search for suitable additional measurement positions or trajectories that allow a higher amount of measurement information to be gained [24, 19].



(a) Fixed-wing aircraft [186] (b) Quadrocopter

Figure 1.2: Examples for sensor-carrying vehicles

In this context, it would be even more beneficial if the sensor vehicles are intended to adapt their movement *online* depending on the current estimate and its uncertainty instead of moving on fixed, precalculated trajectories. Reactions to the outcome of recent measurements would become possible leading to improved further measurement locations and, thereby, to improved process estimates. In this way, a dynamical environment can be observed by a relatively low number of self-adapting robots getting to positions where measurements are highly profitable. Such an approach is an example for a *Cyber Physical System* (CPS) describing the integrations of physical processes with computations with physical processes affecting computations and vice-versa [108]. More specifically, the approach can be considered a *Dynamic Data-Driven Application System* (DDDAS) [41, 42]. The term DDDAS describes systems that directly integrate the collected measurements into the system model and adapt their future measurement process based on the updated model. Several dynamic data-driven approaches have been successfully used for environmental monitoring [1, 118, 134].

However, designing an online approach for monitoring atmospheric dispersion processes is highly challenging since all necessary computations as well as communication have to be performed in real time in practice. Thus, there is a high need for powerful computing platforms and for efficient algorithms including simple, yet precise forecast models that capture the physics of the underlying process. This work sets its focus on the second part: Designing efficient PDE-based dynamic data-driven monitoring approaches to atmospheric dispersion monitoring.

1.3 Related Monitoring Approaches

The application of mobile sensor vehicles in environmental monitoring and, in particular, in atmospheric dispersion monitoring is a large research field. A number of different approaches, concepts and strategies have been proposed, each with its individual hardware constraints, sensor model, process model and adaptation strategy. One of the main classification characteristics is the objective of the monitoring approach. While some approaches have the simple goal to gather measurements, other approaches try to

locate the source of the problem or to track the contaminant cloud. Further approaches are designed to identify process parameters or to estimate the current process state.

Measurement Gathering

Most work dealing with the objective of gathering measurements focuses on the hardware that is applied, i.e. the UAV that are used, the sensors applied for providing measurements or communication devices. Hence, the concerning approaches have not only been developed theoretically, but they have already been tested practically. For example, the vertical distribution of aerosol over the Indian Ocean has been determined by lightweight aircrafts [37], and meteorological data like wind velocities and turbulence [163] or temperature related information [189] have been gathered by a single mini UAV. Other approaches deal with multiple micro UAV or a swarm of UAVs to face tasks in environmental monitoring and disaster response [4, 40] The paths of the UAV are in most approaches precalculated and the vehicles follow the path in an auto-pilot way. Other approaches use teleoperation to control the sensor vehicles.

Contaminant Cloud Tracking

The monitoring goal of another group of monitoring approaches is the tracking of a contaminant cloud. In this task, the boundary of the cloud or predefined concentration level curves is to be estimated over time so that future impacts can be assessed. Besides using qualitative models representing the cloud [52, 22], the boundary of the cloud can, for example, be represented by a specific shape whose parameters have to be estimated [89]. More specifically, polygons [173], splinegons [194, 200, 172], or an underlying grid [164] are used to approximate the level curves. Repeatedly, the positions of the associated vertices are updated using the measurement results by a swarm of UAV. For the mentioned approaches, the paths of the UAV mostly follows the level curve and formation control is enforced. The tracking approach proposed in [55] uses another way applying an optimization-based Model Predictive Control (MPC) scheme where the sensor vehicles are cooperatively moving to positions that are linked with a high error variance. Compared to other monitoring tasks as state estimation, tracking approaches promise a computationally cheap way of estimating a contaminant cloud. However, the dynamics of the process are not considered physically. Moreover, only assessments regarding the boundary or a concrete level set of the contaminant cloud are possible. One does not know anything about the actual amount of concentration at interesting locations.

Source Localization

In other applications, the source location of an accidentally released contaminant plume or cloud might be unknown. Hence, another monitoring task is to find the location of the source. To tackle this challenge, a number of approaches uses biologically inspired strategies [112, 106, 119]. Imitating the behavior of, for example, silkworm moths searching for a pheromone source, sensor vehicles are meant to approach the source location. While these approaches are computationally lightweight, they come along with problems in dynamic environments. Moreover, the sensing and actuation capabilities of the robots differ from the ones of animals, so that these models are stated to be of limited use [16].

A number of other source localization strategies focuses on simple gradient or flow following rules [154, 87, 76, 153]. In these approaches, the chemotaxis strategy, in which the sensor vehicles follow the

concentration gradient, or the anemotaxis strategy, in which the sensor vehicles are moving in upwind direction, are applied. A related approach [121] guides the sensor vehicles in direction of the neighbor measuring the highest concentration. Again, all these approaches are very simple but have problems in high dynamic environments. More sophisticated approaches rely on the minimization of an information theoretic measure considering vehicle dynamics [116]. Related solution methods for the localization strategy can also be found in literature concerning static sensor networks. Moreover, work has been presented covering the question how many sensors should be used and how they should be deployed [6], whereas in a power policy switching on and off the sensor nodes is proposed in [43] . The sampling technique presented in [128] is based on an analytical solution of the considered process equation.

Parameter Estimation

Boundary tracking and source localization both can be considered as special cases of parameter estimation. Research in the area of parameter estimation related to optimal sensor placement has been carried out since the 1970s [102]. The work mainly concentrated on static sensor networks and simple one-dimensional problems. Strategies for mobile sensor networks and the development of optimal sensor trajectories were proposed starting in the late 1980s. While an approach for a single sensor vehicle has been considered in [31], an approach suited for a group of sensor vehicles has been developed in [142]. An overview of the early work in this field can be found in [187]. Basis of the monitoring approaches is the representation of the monitored system in form of a Distributed Parameter System (DPS), e.g. in form of a PDE. The sensor vehicle trajectory problem can be formulated as an optimal control problem. The objective function consisting of a weighted sum of the control energy and the mean square estimation error at final time has to be minimized. A suitable measure for the estimation error, or, in other words, the quality of the parameter estimate, is the Fisher Information Matrix (FIM). According to the Cramer-Rao inequality, the inverse of the FIM represents a lower bound of the error covariance matrix for the estimates [191]. Thus, the goal is to maximize a scalar measure of the FIM. The challenge is that the FIM and, thus, the objective function depends on the parameters to be estimated. This means that the optimal parameters have to be known to be able to calculate optimal trajectories for the sensor vehicles. First methods to tackle these challenges have been proposed in [187, 188]. Another method to address them is the use of a closed-loop-controller. Based on the measured data, the parameter estimates are improved and the trajectories can be adapted accordingly. In [162], first steps towards such a closed-loop control are presented. Furthermore, realistic vehicle dynamics are considered and the optimal control function is minimized subject to the vehicle constraints. Similar techniques also include actuators [181] together with sensors or extend the solution methods to heterogeneous problems [182]. While these methods provide optimal trajectories in relation to the objective function that is minimized, a huge drawback of theses methods is that they are computationally expensive and an application in real-time scenarios is at least questionable. Thus, it is desirable to implement simplifications to theses approaches. An exemplary simplification is the parametrization of the trajectories based on B-splines [202]. In this way, the space of design variables can be reduced drastically and more efficient computation and parallelization is enabled.

Other approaches avoid the use of a DPS and focus on closed form analytical models. While this allows a computationally much cheaper way, modeling assumptions have to be fulfilled or model inaccuracies have to be accepted. Exemplary approaches are based on the Gaussian Plume model [35, 105] or on a puff model [160].

State Estimation

In many cases, not only parameters but especially the state of the system is unknown and has to be estimated. Unknown initial conditions as well as uncertain process parameters and model noise prohibit exact knowledge of the process state. Thus, another type of monitoring application is aimed to estimate the current process state, e.g. the amount of concentration at various positions of the problem domain.

Identification of measurement locations

The development of monitoring strategies for state estimation was mainly driven by the Numerical Weather Prediction (NWP) community in the 90s and early 2000s. It was investigated where sensors of a static sensor network should be preferably placed so that the uncertainty of a weather forecast at a specific time based on the provided measurements could be reduced. To this end, a number of approaches are sensitivity-based. The gradient or adjoint method uses the adjoint equations of the system to identify regions with the largest sensitivities, i.e. regions that contribute most to the forecast uncertainty [107, 39]. In contrast, the quasi-inverse linear method considers the sensitivity of forecast errors to initial conditions so that the forecast difference over the verification area is tracked back to its corresponding initial error [140]. A Monte Carlo method is used by the ensemble spread technique to reveal sensitivities. Several different state vectors are propagated in time and locations with the largest deviations between the forecasted states are identified [114]. Another alternative is the singular vectors technique, which uses measures depending on the singular vectors of the system for the identification of additional measurement locations [24, 25, 133]. For a high-dimensional system, typical for the considered applications, sensitivity-based methods provide a relatively simple means to identify potentially profitable measurement locations. Nevertheless, not only is the actual application of sensor vehicles excluded, but also the observation uncertainty is not taken into account. As an alternative to the sensitivity-based approaches, the Ensemble Transform Kalman Filter [19, 117] can be used to minimize forecast uncertainty represented by the error covariance matrix. An ensemble of states is propagated in time following the principles of the Ensemble Kalman Filter (EnKF) [57] and a transformation matrix is applied to take the observations into account. The transformation matrix represents a given sequence of possible waypoints of the sensor vehicles so that a sequence minimizing the forecast error variance can be chosen.

Considering vehicle dynamics

Nonetheless, the upper approaches all do not entirely include the guidance and control of the sensor vehicles and put more emphasis on the identification of discrete measurement locations than on realistic vehicle trajectories. Trajectory learning to guide the vehicles from one point to another has been proposed as a solution method [151] as well as the solution of a Mixed Integer Linear Program (MILP) modeling the reduction of a predefined uncertainty function subject to vehicle dynamics and collision avoidance [198]. On the other side, apart from the NWP community, new approaches have been proposed in the last years that are all based on an optimal control problem to find optimal trajectories that minimize a scalar measure of the error covariance matrix. The approach proposed in [28] sets up an optimal control problem that is based on the minimization of the error variance. Such an optimal control problem is also the basis of the work presented in [34], which uses an information theoretic measure, the mutual information, as a performance measure in the objective function. Furthermore, simplifications to avoid the solution of the greedy optimal control problem are proposed, e.g. gradient climbing. Both of the aforementioned approaches still do not consider the actual dynamics of the sensor vehicles. In

contrast, [82] uses the trace of the error covariance matrix provided by the use of the Extended Kalman Filter as objective function to be minimized and solves this problem subject to realistic vehicle dynamics. The consideration of realistic vehicle dynamics is also an important part of the work presented in [199]. Similar to the presented works in parameter estimation, a receding horizon strategy is used here so that the trajectories are adapted online according to changes in the state estimates.

Simplifications for online approaches

Solving the optimal control problem subject to process dynamics, uncertainty dynamics and vehicle dynamics is challenging considering the real-time constraints of practical online dynamic data-driven applications. Simplifications have to be introduced to make the solution of the complex, coupled problem computationally tractable. The solution approaches proposed in [199] are a gradient based solution technique as well as reduced uncertainty dynamics. Reduced uncertainty dynamics are also used in [75] proposing a similar receding horizon optimal control problem subject to vehicle dynamics and collision avoidance.

Despite the consideration of more efficient solution techniques or simplifications in the propagation of the error covariance, the optimal control problem still might be computationally intractable to be solved in a short amount of time. Simplifications that lead towards a real-time applicable scheme and still guide the sensor vehicle to sufficiently profitable measurement locations have to be developed. A potential solution is the application of projection-based model order reduction methods. The full order model is projected on a lower order model that stills captures most of the dynamics of the original problem. A popular model order reduction method is Proper Orthogonal Decomposition (POD), which uses a singular value decomposition of a representative solution set for the projection. An early approach applying POD in context of static sensor placement has been presented in [5]. For mobile sensor vehicles, a dynamic data-driven strategy has been developed [134, 159] considering POD for the estimation part where the coefficients of the reduced model are fitted to the measurements in a least squares sense. Then, hotspots are identified based on the most dominant modes and a smoothed particle hydrodynamics method is used to guide the sensor vehicles to theses hotspots. Extensions of this approach work with a POD-based reduced Kalman Filter in the estimation part [135].

An approach that completely avoids the solution of an optimal control problem has been proposed in [44]. Instead, an a priori guidance law is set up, which relies on a control-Lyapunov function that is based on the estimation error. As the guidance law is known a priori, no complex computations have to be processed and the approach becomes real-time capable. An extension of the work also includes vehicle dynamics and collision avoidance [45]. Further work in this context focuses on a more lightweight filtering part introducing a Luenberger observer together with an adaptive grid refinement to increase efficiency [46, 65]. The drawback of the presented method is that process and measurement noise are not considered in the guidance scheme and that an extension to more sophisticated problem scenarios is not straight forward. Furthermore, due to the a priori guidance law, the sensor vehicle is not able to adapt to dynamically changing conditions.

Decentralized approaches

The aforementioned approaches (except for [44], which is proposed to be also applied in a decentralized scenario) can be characterized to be centralized. This means that the information gathered at the sensor nodes is transmitted to a central station where all the information is present. A central supercomputer is in charge of processing every computation job and the necessary information, e.g. the control inputs, are sent back to the sensor vehicles. Such a system has high communication requirements, is not scalable,

has a central point of failure and is potentially not fast enough to cope with rapid changing dynamics. Hence, the introduction of a decentralized approach seems to be desirable. In such a system, a single sensor vehicle unit possesses only little knowledge of the overall system. By communicating with its neighbors in its local vicinity and performing measuring tasks, new local information become available being processed locally on-board the sensor vehicle. An omniscient central node is completely avoided. However, when shifting the computational load from the central node to several distributed nodes, it has to be noted that the on-board computational power is much less than the one of the central supercomputer. This is why one even more has to keep an eye on the simplicity of the computations.

Besides the requirement of computational efficiency, decentralized coordination algorithms have to be developed to attain a cooperative motion of the sensor vehicles. An overview of this topic is given in [120, 26], which sets the focus on keeping a specific pattern of the network. A decentralized coordination algorithm for optimal data collection is proposed in [109]. In this approach, an adaptive coverage control is proposed, in which the sensor vehicles move along parametric ellipses that rely on current uncertainty estimates. Besides these works, decentralized monitoring approaches have among others been applied in the fields information gathering [71], detection [190], contour estimation [55] or source localization [121].

The literature on decentralized monitoring for state estimation is sparse. The critical aspect in this context is the process model. If the process is modeled by a PDE model, the resulting discretization consists of an extremely high number of state variables that make the problem computationally intractable to be solved in real-time on a system with low computational power. For this reason, greedy PDE models are avoided and statistical models are used instead. While a strategy that employs Gaussian processes for a team of mobile sensor vehicles has been presented in [170], other works use Gaussian processes for adapting the sampling rate [101] or the Bayesian theory as well as consensus to develop a decentralized state estimation scheme [90]. Further alternatives as artificial potential fields [62] or fuzzy logic models [38] have also been proposed.

1.4 Aim, Contribution and Outline of the Thesis

The aim of this thesis is the development of new model-based monitoring strategies for state and parameter estimation of atmospheric dispersion processes. In particular, the focus is set on the application, modification, extension, combination and new development of higher level algorithms and approaches concerning modeling, simulation, data assimilation, parameter estimation, identification of suitable measurement locations and vehicle control as well as on their coupling. It is concentrated on the proposition and computational evaluation of theoretical approaches, whereas hardware aspects and an experimental implementation are beyond the scope of this work. Instead of considering and modeling a specific test application, the approaches should be designed to fit a general class of dispersion problems with focus on medium range problems on spatial scales between a few meters to several kilometers. For application in more sophisticated problems, the developed methods should be extensible to more specific problems in an easy and straightforward way.

Moreover, a superior aim is to integrate the following highly beneficial properties to large extent in the monitoring approach:

- *Dynamic Data-Driven:* The measurement process is adapted online based on the estimates resulting from prior integrated measurements. Sensor vehicles are guided to locations that provide a high amount of information based on the current estimate and estimate statistics, respectively. In

this way, reactions to changing situations in dynamic processes become possible and the estimate can be improved (see also Section 1.2).

- *PDE-based:* The estimate is based on predictions stemming from a mathematical forecast model of the process. As the forecasts should be physically plausible and precise, partial differential equations (PDEs) should be considered in this context. PDEs provide a good possibility to model relations and evolutions of dispersion processes.
- *Decentralized:* Many data-driven strategies from the state of research contain a central supercomputer. Measurements are sent to the central which combines all measurements and the predicted model information and computes new control inputs that are sent back to the sensor vehicles. Obviously, such approaches harbor the risk that if the central node breaks down, the whole system will break down. Furthermore, they lack scalability and there are high communication requirements. To circumvent these disadvantages and to gain scalability and robustness, decentralized approaches appear to be well-suited for the posed requirements. The central computing node is abandoned and, instead, calculations are performed on-board the sensor platforms. Every unit processes its local information and communicates with its local neighbors to exchange information.

The integration of all properties into the monitoring approach is very challenging and, until now, hardly found in related literature, which either treats precalculated trajectories, relatively simple models or centralized computation topologies (cf. Section 1.3). Especially the combination of the two first properties is difficult. A dynamic data-driven strategy requires real-time computations, whereas the discretization of PDE models usually results in high-dimensional problems whose solution takes a lot of computation time. Also, when dealing with a high number of sensor vehicles or with a large problem domain, it is hard to get computation results in real time. Strategies for simplifying the considered problems have to be applied to create efficient, but still accurate methods. Simplification examples that are applied in this work are the avoidance of complex optimal control problems, use of a smaller model basis or the partitioning of the domain.

The problem described before becomes even more dominant when dealing with decentralized approaches possessing computational units with less capability. Further reduction strategies have to be applied to attain results in a relatively short amount of time. This includes an efficient communication and cooperation among the sensor vehicles so that the total work can be shared equally.

The resulting monitoring strategy is composed of different parts stemming from a wide variety of research fields so that knowledge in different research areas is necessary. Existing methods have to be chosen, applied, modified or extended and new methods have to be developed and coupled with the other parts.

This thesis contributes to the state of the art by proposing in total three new efficient PDE-based dynamic data-driven monitoring approaches, which are the subject of the following chapters:

A New Centralized Approach to PDE-based Dynamic Data-Driven Monitoring (Chapter 2)

The first approach is a new *centralized* PDE-based dynamic data-driven monitoring strategy. A complex optimal control problem, usually considered in related literature for sensor guidance (cf. Section 1.3), is avoided to gain efficiency. Instead, vehicle control is decoupled from the dimension of the process state, and control and estimation are considered separate parts. The advection-diffusion equation is selected as PDE model for the dispersion process and a finite element method (FEM) with characteristic Galerkin method is applied for solving the forecast problem. For efficient data assimilation, an Ensemble Square Root Filter (EnSRF) is employed, using a state-dependent covariance

matrix for model error. This data assimilation method is extended applying state augmentation in order to not only obtain the process state vector but to jointly estimate process state and source output. A new target point generator identifies informative measurement locations in a sequential procedure based on the state estimate's statistics and on probable source locations. The target points serve as input for an optimization-based cooperative feedback vehicle controller that is in charge of guiding the sensor vehicles to the target locations. The approach together with a number of fundamentals and assumptions used throughout this work is presented Chapter 2. This chapter also includes a thorough evaluation of the approach including an examination of the influences of specific method parameters and the scalability of the approach.

A New Decentralized Approach to PDE-based Dynamic Data-Driven Monitoring (Chapter 3)

Based on the centralized approach, the second approach is a novel *decentralized* dynamic data-driven approach working with forecasts stemming from the discretization of a PDE model. Compared to the centralized approach, all the calculations are performed on-board the sensor vehicles and information is exchanged between neighboring sensor vehicles that are located within a specific, limited communication range. The particular challenges in this approach are the limited on-board computational capability and the limited communication ranges leading to different estimates at different vehicles complicating data fusion. To compensate for the limited computational capabilities, Proper Orthogonal Decomposition (POD) and Galerkin projection are used to largely reduce the problem dimension. POD requires several simulation runs with exemplary parameter settings, initial conditions, etc. prior to the actual application so that a new set of basis vectors can be constructed from the precalculated data. Suitable propositions for the choice of different offline simulation runs are provided. Prediction as well as data assimilation are working with the reduced order model making the approach much more efficient. The data assimilation part has to be adapted to the decentralized topology and consists of a two-step procedure. The Kalman Filter is used to integrate the own measurements and Covariance Intersection is used to fuse local data. An existing decentralized variant of the vehicle controller is employed and supplied with target points obtained from an extension of the previously presented target point generator. Chapter 3 provides a detailed description of this monitoring approach and shows its working with a number of test cases.

A New Partitioned Approach to Decentralized PDE-based Dynamic Data-Driven Monitoring (Chapter 4)

The third approach further pursues decentralization and represents, to the best of the author's knowledge, the first *partitioned* PDE-based dynamic-data driven monitoring approach. To increase scalability, the problem domain is subdivided into several subdomains and every sensor vehicle is assigned to the subdomain it is currently located in. Instead of maintaining a model of the whole problem domain, only calculations concerning the local subdomain have to be performed. Based on concepts from Domain Decomposition, the sensor vehicles exchange information with vehicles in neighboring parts concerning the forecast. For data assimilation, a novel partitioned variant of an Ensemble Square Root Filter is developed that allows an efficient exchange of information between vehicles on neighboring domains. Target generation and vehicle control are extended in such a way that it is possible for a sensor vehicle to go from its current subdomain to a neighboring subdomain where the need of an additional sensor device is higher. In this way, sensor vehicles can dynamically adapt their distribution to changing conditions, which are likely to occur in dynamical processes. The partitioned approach as well as its evaluation and comparison to the other approaches is subject of Chapter 4.

Chapters 2, 3 and 4 present and evaluate the three new monitoring strategies. Every strategy consists of the same structure concerning its building blocks so that the outline of every chapter is similar. Chapter 2, however, additionally contains a number of basics and the statement of assumptions that are also used in later chapters. The work is concluded by a summary and an outlook in Chapter 5.

2 A New Centralized Approach to PDE-based Dynamic Data-Driven Monitoring

This chapter presents a new centralized dynamic data-driven approach to atmospheric dispersion monitoring. The proposed approach is based on a discretized PDE model enabling physically realistic forecasts and is, nevertheless, designed to save computational time keeping possible real-time requirements of related practical approaches in mind. Instead of solving a sophisticated optimal control problem subject to full process, vehicle and uncertainty dynamics to obtain optimal vehicle trajectories, a suboptimal solution structure is considered consisting of, among others, an ensemble-based data assimilation method, an iterative uncertainty-based identification method for suitable measurement locations and a cooperative vehicle controller.

Aspects of this chapter have been previously published in [147] and [56]. In [147], a first variant of the general structure of the monitoring approach for state estimation is presented including the description of its individual parts: forecast, data assimilation, target point generation and vehicle controller. An extended variant is subject of the approach presented in [56] considering continuous measurements and introducing local appealing points (see Subsection 2.5.2). Compared to these publications, this chapter also contains the proposition of a joint state and parameter estimation approach for additionally estimating source locations and another type of ensemble filter is applied for data assimilation. Furthermore, the handling of a model error covariance matrix depending on the current process state is investigated and a more detailed description of the monitoring approach, especially regarding modeling and data assimilation, and a more thorough evaluation of the approach is provided.

After a short discussion of related centralized monitoring approaches and an overview of the proposed monitoring approach in Section 2.1, the multiple building blocks of the new approach are presented in Sections 2.2 - 2.6. Moreover, a number of basics are described in these sections that will not only form the basis of the strategy described in this chapter but also for further extensions presented in Chapters 3 and 4. The approach is evaluated and tested in Section 2.7 and concluded with some discussions on benefits and drawbacks of the approach in Section 2.8.

2.1 A New Centralized Monitoring Strategy: Contribution and Overview

The main goal of the monitoring approach presented in this chapter is the online estimation of the current process state as well as of current process parameters for atmospheric dispersion processes. In order to improve the estimates, a network of multiple mobile sensor vehicles is applied to gather measurements that can be combined with forecasts stemming from discretized PDE process models.

Related work in this context has been presented in Section 1.3. The approaches proposed in [142, 187, 187, 188, 162, 182, 82, 199] generally result in the task of solving an optimal control problem to obtain suitable control inputs for the sensor vehicles. A scalar measure on the error covariance matrix or on the FIM has to be minimized subject to vehicle dynamics, covariance evolution, measurement aspects and potentially other criteria. The solution of such an optimal control problem is costly so that simplification approaches like the parametrization of the trajectories [202] or reduced uncertainty dynamics [199] have been proposed. However, for the mentioned approaches, the computation time is not an essential factor

since these approaches are generally intended to find optimal vehicle trajectories prior to their actual application.

Typically, the process state and parameter vector are not known a priori so that the quality of the pre-calculated trajectories might be much worse than originally assumed. Hence, it is suitable to adapt the sensor vehicle movement in an online manner based on the current estimate and the measurements taken so far resulting in a DDDAS. These systems require computation results to be available in real time and prohibit the use of sophisticated optimal control problems. Only a few approaches have been proposed tackling this challenging problem. Sophisticated optimal control problems have to be avoided and the estimation part and controller part are split. While smooth particle hydrodynamics are used in [134] for vehicle control and model order reduction for a performance improvement, simple Lyapunov-guidance schemes complemented with an adaptive grid refinement have been proposed by [46, 65].

In this chapter, a new dynamic data-driven PDE-based approach is proposed that avoids the solution of a complex optimal control problem for vehicle guidance. To increase the efficiency and advance towards real-time applicability, a sub-optimal method is derived being based on the interaction of a process simulation, a data assimilation, a target point identification and a vehicle control part. The proposed method can be regarded to be dynamic data-driven, capable of producing physically precise forecasts (due to the use of PDE model), efficient (sub-optimal problem structure, use of ensembles) and cooperative (vehicle controller). Also, the approach can be characterized to be centralized, i.e. a central supercomputer is in charge of performing almost all necessary calculations. This is in contrast to Chapters 3 and 4, in which decentralized monitoring approaches are presented.

To develop the monitoring approach, several building blocks are required. These building blocks comprise a number of existing and partially well-established methods from different research areas that have to be selected thoroughly and have to be applied and potentially modified adequately to fit the present requirements. Newly proposed methods complement the approach and suitable coupling strategies between the several parts are designed to enable an effective and fruitful interaction. An overview of the resulting strategy is depicted in Figure 2.1 and described in the following giving the outline of this chapter.

Assuming an observation is available at a specific time, the prior process estimates have to be forecasted to observation time. This forecast is performed based on a PDE model of the dispersion process – in this work the advection-diffusion equation. As solution method, the finite element method complemented with a characteristic Galerkin approach is applied. Section 2.2 contains details about modeling and forecasting addressing issues of how a general atmospheric dispersion process could best be modeled, which model assumptions are justified and how an FEM solution can be obtained.

The sensor vehicles, whose dynamics, observation model, and topology are described in Section 2.3, send their measurements to the central. At this point, the model forecasts are combined with the measurements obtained from every sensor vehicle. The combination procedure, also called data assimilation, is the subject of Section 2.4. Based on the description of the Kalman Filter, a standard method in data assimilation, the Ensemble Square Root Filter (EnSRF), applied in this work is derived and important assumptions and properties concerning involved errors are stated. The considered assimilation method is extended so that process state and process parameters, in particular the source output function and location, can be estimated jointly. It is further explained, how the ensemble is set up and maintained during the forecasts. In particular, a new form for an efficient handling of a state-dependent model error covariance matrix is proposed.

The data assimilation method not only processes the mean of the process estimate, but also the error covariance matrix, a measure of the uncertainty of the mean estimate. For this reason, the error covariance

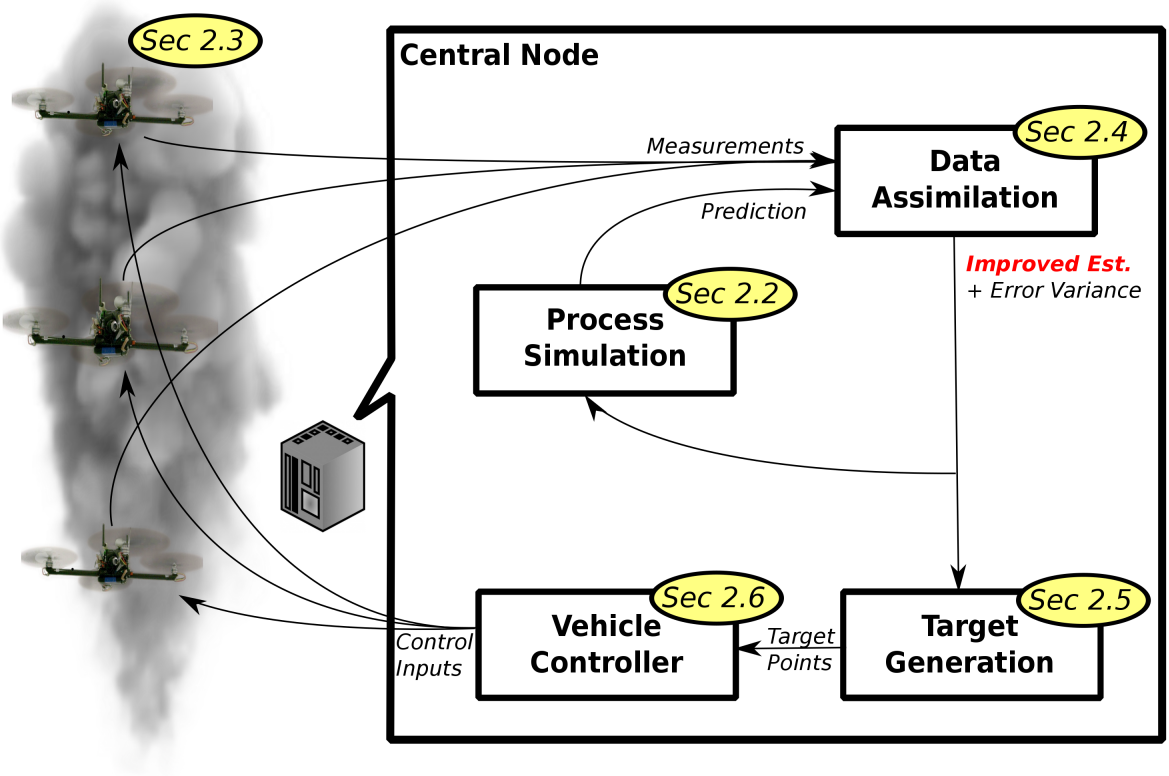


Figure 2.1: Overview of the proposed centralized monitoring approach

matrix is a good means to determine locations that seem to be useful for additional measurements. A new sequential procedure to identify new informative measurement locations based on the error covariance matrix as well as on potential solutions of the source is proposed and described in Section 2.5. As output of the proposed procedure, one obtains a number of target points that should be reached by the sensor vehicles.

In the final step of this cycle, these target points are handed to a vehicle controller, which is in charge of guiding the sensor vehicles to the prescribed target points in a cooperative way. To achieve this, an optimization-based feedback controller based on a Mixed Integer Linear Program and on an MLD formulation of the problem is employed (see Section 2.6). The resulting control input is sent back to the sensor vehicles, which adapt their movement accordingly.

2.2 Modeling and Simulation of Atmospheric Dispersion Processes

A mathematical model is required to describe and predict the considered atmospheric dispersion process. The model should describe the behavior of the pollutant being emitted from sources and dispersing in the air. This is typically a physically very complex process so that a number of different models have been proposed in the past, each being based on different assumptions and application scenarios. Approximately 100 different atmospheric dispersion models are listed by the free online encyclopedia Wikipedia [195]. The choice of the model depends on the considered scenario as well as on individual demands like desired accuracy and available computational capabilities.

Roughly, one can distinguish two general types of atmospheric dispersion models. The first type is based on a purely stochastic or statical representation of the dispersion process. A general review of such kind of methods can be found in [122]. While these methods are rather simple and results can be obtained in a short amount of time, they do not consider the actual physical relations and dynamics of the underlying process. This can lead to unphysical forecasts and dynamically changing properties cannot be represented physically. For this reason, the focus is set on models that are based on the physical characteristics of the atmospheric dispersion process. A comprehensive overview of this kind of models is provided by [74]. The central equation of those models is the advection-diffusion equation, a partial differential equation that describes the dispersion of the pollutant considering diffusion, advection, and source effects. It can be applied together with complementing equations modeling fluid dynamics, chemical reactions or humidity effects, which might also have an impact on the pollutant transport regarding a possibly changing velocity field, varying diffusion coefficients or altered pollutant properties. A number of different classes of solution methods exist to solve the resulting problem. Most of the existing atmospheric dispersion models [195] are based on Gaussian plume models, which rely on analytical steady-state solutions of the advection-diffusion equation requiring a number of strong model simplifications [20, 174, 36, 167]. On the other hand, there are a few models providing precise forecasts by considering a number of different additional physical effects [85, 123], which are, however, not suitable for the real-time applications considered in this work. In order to get precise results in an adequate amount of time, a middle course between both dispersion model types has to be chosen. A Eulerian solution method based on a few model assumptions is implemented in this work using FEM with a characteristic Galerkin method. The own implementation allows a better adjustment of the model to the considered processes and also enables model modifications regarding interactions with the other parts of the monitoring approach.

In the following, physical characteristics of atmospheric dispersion processes are described and the advection-diffusion equation is derived according to [167, 185, 11]. A number of assumptions that help to simplify the resulting equation in the context of this work are illuminated so that the solution process can become more efficient. After a short review of possible other solution methods, the method used in this work, FEM with characteristic Galerkin, is presented.

2.2.1 Physical Characteristics and the Advection-Diffusion Equation

Pollutants are emitted from specific sources, are transported in the air and a potentially progressively expanding pollutant cloud is developing. Which physical effects are of importance for such a scenario? In general, atmospheric dispersion processes are governed by mainly five physical effects (see Figure 2.2):

- *Advection* is the effect that the pollutant is transported with the mean wind velocity.
- *Diffusion* occurs due to the turbulent eddy motion and effects a mixing of the pollutant with the ambient unpolluted air.
- *Deposition* describes the effect of gravitation on the pollutants. Particles settle in gravitational direction.
- *Reaction* describes the interaction between the involved substances. Depending on the amounts of each substance at hand, substances are produced or reduced.
- *Sources* represent the emission of the pollutant at specific locations. They describe where and how much of a specific substance is produced or reduced.

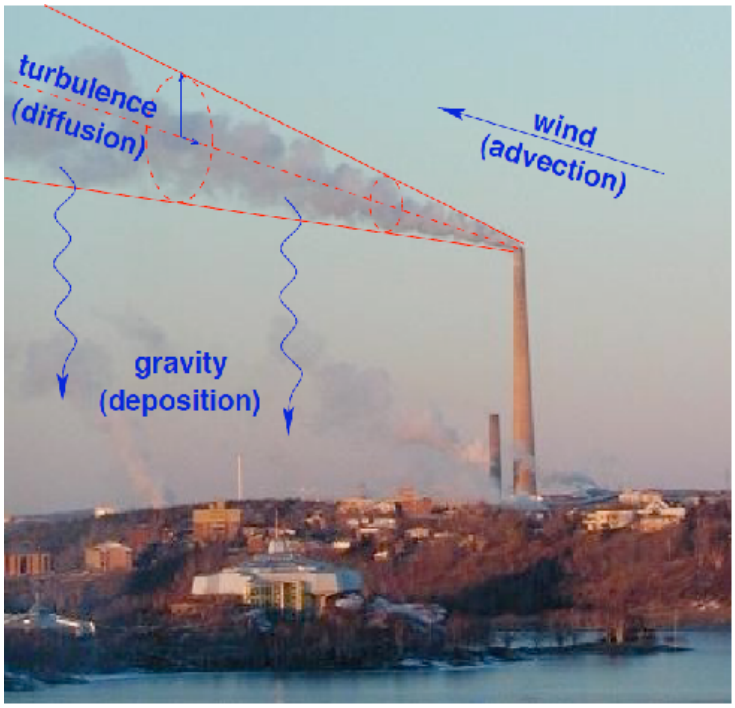


Figure 2.2: Illustration of the main contribution factors to atmospheric dispersion, extracted from [167]

The transport of a quantity due to these factors can be modeled by the so-called advection-diffusion equation, which is derived in the following. The mass concentration of a single substance at the three dimensional location $\mathbf{r} = (r_1, r_2, r_3)^T \in \Omega \subset \mathbb{R}^3$ and time $t \in \mathbb{R}_0^+$ is described by the function $x(\mathbf{r}, t)$. For this entity, the conservation of mass is valid, i.e.

$$\frac{\partial x}{\partial t} + \nabla \cdot \mathbf{j} = s, \tag{2.1}$$

with the nabla operator $\nabla = \left(\frac{\partial}{\partial r_1}, \frac{\partial}{\partial r_2}, \frac{\partial}{\partial r_3} \right)^T$ and the source function $s(x(\mathbf{r}, t), \mathbf{r}, t)$ depending on the concentration $x(\mathbf{r}, t)$ as well as on the location $\mathbf{r}$ and time t. According to (2.1), the source term $s(x(\mathbf{r}, t), \mathbf{r}, t)$ comprises the effects of sources, reaction and deposition. The mass flux $\mathbf{j}(x(\mathbf{r}, t), \mathbf{r}, t)$ contains the contributions of the diffusive and advective effects of the system: $\mathbf{j} = \mathbf{j_D} + \mathbf{j_A}$. The diffusive flux $\mathbf{j_D}$ is assumed to follow Fick's law, i.e. $\mathbf{j_D} = -\mathbf{D}\nabla x$. The negative sign ensures that the substance flows from regions with high concentrations to regions with low concentration. Furthermore, the diffusion coefficient $\mathbf{D}(\mathbf{r}, t)$ is a diagonal matrix, whose entries are typically the turbulent eddy diffusivities. The advective flux is caused by the wind velocity so that it can be expressed with the help of the wind velocity $\mathbf{w}(\mathbf{r}, t) = (w_1, w_2, w_3)^T$ as $\mathbf{j_A} = x\mathbf{w}$.

Inserting the fluxes into Equation (2.1), the advection-diffusion equation reads as follows

$$\frac{\partial x}{\partial t} + \nabla \cdot (x\mathbf{w}) - \nabla \cdot (\mathbf{D}\nabla x) = s. \tag{2.2}$$

While the advection is represented by the second term, diffusion effects are included in the third term and source effects as well as reaction and deposition are contained in the source function $s(x, \mathbf{r}, t)$ on the right-hand side.

2.2.2 Model Assumptions

The advection-diffusion equation (2.2) is not only capable to model atmospheric dispersion processes, but can be also used to describe other transport processes, for example underwater dispersion or solute transport in soils. Thus, the methods that are developed in the following can also be applied to related problem scenarios in other applications.

Nevertheless, it is necessary to make a few assumptions that, on the one side, specialize the equation for a specific class of problem scenarios and, on the other side, simplify the problem with respect to an efficient computational solvability. It is focused on dispersion processes taking place near the pollutant source in the lower troposphere, especially in the atmospheric boundary layer, which are related to much smaller physical and temporal scales [171]. For the remainder of the work, the following assumptions are made:

- *Two-dimensional domain:* The probably most severe simplification of this work is that only two-dimensional scenarios are considered, $\Omega \subset \mathbb{R}^2$. Keeping in mind that realistic applications have to be necessarily modeled in three spatial dimensions, this assumption is made to tighten the focus onto a more comprehensive development of the methodology for data assimilation, adaptive observations, and vehicle control. All methods and approaches developed in this work can be extended to the three-dimensional case in a straightforward way. Only the solvers for the process and vehicle simulation have to be adapted respecting the associated increase in state dimension related to higher computation times.
- *One chemical substance, no reaction or deposition:* It is assumed that only one chemical substance is considered so that only one advection-diffusion equation has to be solved. Furthermore, the deposition and reaction part do not have a large effect on the transport of the considered quantity so that the source term $s(\mathbf{r}, t)$ is independent of the concentration $x(\mathbf{r}, t)$. One may justify this assumption by stating that the mass of the considered particles or gas is very low and that no reaction takes place. Once more, an extension to a scenario that involves deposition or reaction effects, is straight forward as one simply has to add the deposition and reaction effect to the considered equation and has to solve the advection-diffusion-reaction equation instead.
- *Known wind effects:* The wind velocity $\mathbf{w}(\mathbf{r}, t)$ and the turbulent eddy motion $\mathbf{D}(\mathbf{r}, t)$ are known. This means that there is no need to solve additional equations concerning the fluid flow, which strongly facilitates the computation. In practice, the fluid flow might be known approximately, but not exactly. An additional velocity parameter estimation (see Subsection 2.4.4) can alleviate this problem, but is not explicitly considered in this work.
- *Incompressible fluid flow:* Another assumption concerning the background velocity is the incompressibility of the considered fluid flow. An incompressible fluid flow is characterized by the relation $\nabla \mathbf{w} = 0$ so that the second term in (2.2) can be rewritten

$$\nabla \cdot (x\mathbf{w}) = \mathbf{w} \cdot \nabla x(\mathbf{r}, t) \tag{2.3}$$

  This assumption is justified since the Mach-number describing the ratio between the typical flow speed u and the speed of sound a is smaller than 0.3.
- *Homogeneous, constant diffusivity*: For reasons of simplicity, the diffusivity $\mathbf{D}$ is assumed to be homogeneous and constant. With this, the diffusion matrix $\mathbf{D}(\mathbf{r}, t)$ can be replaced by the scalar diffusivity D.

- *Process dominated by advection*: The considered processes excel by a high Péclet number

$$\mathrm{Pe} = \frac{lw}{D}. \tag{2.4}$$

  This means that the characteristic length l multiplied by a characteristic speed w is larger than the diffusivity, i.e. the process mainly disperses due to advective wind velocity. The assumption aggravates the work with FEM (see Subsection 2.2.3), but has advantages when applying domain decomposition methods (see Chapter 4).
- *Source might be unknown*: The source term s on the right-hand side might be time-varying and can be known or unknown. If the source is unknown, one has to additionally estimate the source term (see Subsection 2.4.4) with respect to space and time.

Inserting the assumptions in (2.2), the simplified advection-diffusion equation considered in the remainder of this work reads

$$\frac{\partial x}{\partial t} + \mathbf{w} \cdot \nabla x - D\Delta x = s \tag{2.5}$$

with the Laplace operator $\Delta x = \nabla \cdot \nabla x$.

2.2.3 Solving the Advection-Diffusion Equation

To solve (2.2) or its simplified form (2.5), a number of different solution approaches are available. They can be mainly categorized in analytical Gaussian Plume models, Lagrangian approaches, and Eulerian approaches. A comprehensive review can be found in [78].

Gaussian Plume Models

One of the probably easiest possibilities is the use of an analytical solution of the advection-diffusion equation. So-called Gaussian Plume Models provide such a possibility. Introduced in the 1930s [20, 174] and repeatedly refined in following decades [167], Gaussian Plume Models, e.g. the popular AERMOD model [36], are especially applied in the field of emission calculation for industrial plants [17]. While analytical solutions are quick and computationally lightweight, a number of further simplifications have to be made so that the PDE is analytically solvable [167]. Amongst several other assumptions, it has to be assumed that the contaminant is emitted at a constant rate and that the wind velocity is constant. Hence, compared to the assumptions made in Section 2.2.2, the assumptions are more severe, and, what is more, cannot be resolved by a straightforward extension. For this reason, Gaussian Plume Models are not considered within the scope of this work and a more sophisticated technique is necessary.

Lagrangian Models

Lagrangian methods [196] are based on a large set of small parcels called particles whose trajectory is calculated individually according to the advection-diffusion process. The concentration field can be calculated from the distribution or density of the particles. Thus, a change in global concentration corresponds to a displacement of a single particle. The basis for the work in this field was laid by Taylor in 1922 [175] and criteria for a theoretically correct model were derived in [177]. Today, many different

models are applied in practice [168, 84, 51, 156]. Amongst these methods, one also finds the Gaussian puff method [84, 51, 156], which reduces the number of particles, i.e. the computational cost, and describes the diffusion with Gaussian puffs whose centroids are moved with wind velocity to represent advection. Using Lagrangian models, the computation is usually not very costly. Moreover, Lagrangian models are easy to understand and to implement. However, many parameters and relations have to be determined in an empiric or semi-empiric way. Furthermore, it is difficult to include other physical aspects, e.g. sophisticated reaction terms. Even more severe is the drawback that the transfer of the output of the Lagrangian model to the fixed grid, which is related to further information, e.g. background velocity, is expensive and might lead to errors.

Eulerian Models

The third type of methods to tackle the advection-diffusion equation is the Eulerian approach. In Eulerian Models, the model domain is divided into a fixed array of grid cells and the advection-diffusion equation (2.2) is solved on the defined grid. A number of numerical techniques like the finite difference, finite volume or finite element method are available to solve the resulting problem. Example applications [85, 123] illustrate that besides the advection-diffusion equation, a lot of further calculations can be performed on the grid, for example fluid flow, turbulence, humidity or complex reactions equations. Thus, Eulerian Models can be arbitrarily refined and sophisticated depending on their application and their desired accuracy. Although one has to note that the cost of Eulerian Models depending on their sophistication is relatively high and numerical diffusion can be introduced due to the advection calculation, the Eulerian model seems to be the right choice for this work. Eulerian approaches are capable of treating complex processes on complex terrain and do not rely on empirical relations and parameters as Lagrangian models. Moreover, most of related data, like meteorological information, measurements, sensor positions, etc., corresponds to a fixed spatial grid. Hence, it is desirable to also have a model working on the same grid avoiding the need of repeatedly transferring data from the Lagrangian space to the fixed grid. Model reduction methods, as used in Chapter 3, are also tailored for Eulerian methods and, applying them, the model forecasts can be speeded up and have the potential to outperform cheap Lagrangian methods.

The Finite Element Method with Characteristic Galerkin

For the aforementioned reasons, Eulerian models provide a good mean to solve the advection-diffusion equation. A popular Eulerian method is the finite element method (FEM) that is applied within the scope of this work. This method is a standard numerical technique to solve partial differential equations. It is based on a transfer of the partial differential equation into a system of ordinary differential equations that can be solved numerically. To achieve this goal, the problem domain is divided into a number of elements, over which the PDE is integrated locally. The overall solution system is obtained by assembling the elemental integrals.

FEM can be applied in scenarios related to complex problem domains. It is widely used in practice, especially in the structural mechanics community where it was developed. For fluid dynamic problems and advection-diffusion problems, the method can also be used [69]. However, problems might occur for highly advection-dominated flows, which are considered in the context of this work (see Subsection 2.2.2).

An important indicator for advection dominated flows is a high cell Péclet number.

$$\mathrm{Pe}_{\mathrm{cell}} = \frac{\Delta r w}{D}. \tag{2.6}$$

The cell Péclet number is, like its global counterpart (2.4), a dimensionless quantity which describes the ratio between the grid spacing Δr multiplied by a characteristic speed w and a characteristic diffusion coefficient D. For high cell Péclet numbers, spurious oscillations can arise leading to instabilities. According to Equation (2.6), the grid spacing Δr hast to be reduced in order to reduce the cell Péclet number. However, the computational effort grows significantly with a higher problem dimension so that stabilizing methods for the FEM approach should be considered. There are a number of different approaches to solve this issue as for example Streamline-Upwind-Petrov-Galerkin [23] and Taylor-Galerkin methods [49].

In this work, a characteristic Galerkin method is applied [138, 94, 33]. Compared to the standard finite element method, the discretization of the advection part is treated in a different way and the method of characteristics is applied to track the advective movement of particles residing at specific locations from one instance of time to another. In this way, non-oscillatory solutions without numerical diffusion can be expected.

In the following, the characteristic Galerkin finite element method for the advection-diffusion equation (2.5) is derived. The initial boundary value problem to be solved reads

$$\frac{\partial x}{\partial t} + \mathbf{w} \cdot \nabla x - D\Delta x = s \quad \text{in } \Omega \tag{2.7a}$$

$$x = x_s \quad \text{on } \partial\Omega_D \tag{2.7b}$$

$$\nabla x \cdot \mathbf{n} = t_s \quad \text{on } \partial\Omega_N \tag{2.7c}$$

$$x(\mathbf{r}, 0) = x^{(0)}(\mathbf{r}) \quad \text{in } \Omega \tag{2.7d}$$

with the Dirichlet boundary condition (2.7b) on boundary $\partial\Omega_D$ and the Neumann boundary condition (2.7c) with outpointing normal vector $\mathbf{n}$ on boundary $\partial\Omega_N$, where $\partial\Omega_D \cup \partial\Omega_N = \partial\Omega$. Additionally, the initial condition (2.7d) provides the concentration $x^{(0)}(\mathbf{r})$ at time $t = 0$.

When applying the characteristic Galerkin method, the total derivative of the concentration

$$\frac{\mathrm{D}x}{\mathrm{D}t} = \frac{\partial x}{\partial t} + \mathbf{w} \cdot \nabla x \tag{2.8}$$

plays an important role since it describes the temporal change a particle undergoes if it is transported with flow velocity $\mathbf{w}$. Inserting the total derivative into the advection-diffusion equation (2.7a) yields

$$\frac{\mathrm{D}x}{\mathrm{D}t} - D\Delta x = s. \tag{2.9}$$

The next step consists of time discretization. Between starting time $t = 0$ and final time $t = T$, the time scale is divided into n_T equidistant segments with size Δt resulting in the time stages

$$0 = t^{(0)} < t^{(1)} < \ldots < t^{(k)} < t^{(k+1)} < \ldots < t^{(n_T)} = T. \tag{2.10}$$

For simplicity, the notation $x^{(k)} = x(t^{(k)})$ is used for all entities in the remainder of this work. The total derivative in (2.9) can be discretized by the use of the characteristics $\mathbf{r_c}$, which describe the trajectories along which the particles are transported with velocity $\mathbf{w}$, i.e.

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{r_c}(\mathbf{r},\tau;t) = \mathbf{w}(\mathbf{r_c}(\mathbf{r},\tau;t),t) \tag{2.11}$$

$$\mathbf{r_c}(\mathbf{r},\tau;\tau) = \mathbf{r}. \tag{2.12}$$

According to this definition, a particle located at position $\mathbf{r}$ at time $t^{(k+1)}$ regarding advection was formerly located at position $\mathbf{r_c}(\mathbf{r}, t^{(k+1)}; t^{(k)})$ at time $t^{(k)}$. Following this approach, the total derivative can be discretized as

$$\frac{\mathrm{D}x}{\mathrm{D}t} \approx \frac{x^{(k+1)}(\mathbf{r}) - x^{(k)}(\mathbf{r_c}(\mathbf{r}, t^{(k+1)}; t^{(k)}))}{\Delta t}. \tag{2.13}$$

With the aid of the implicit Euler time integration method, the time discretized version of Equation (2.9) yields

$$\frac{x^{(k+1)}(\mathbf{r}) - x^{(k)}(\mathbf{r_c}(\mathbf{r}, t^{(k+1)}; t^{(k)}))}{\Delta t} - D\Delta x^{(k+1)}(\mathbf{r}) = s^{(k+1)}(\mathbf{r}). \tag{2.14}$$

To solve (2.14), the finite element method is applied to the specific problem according to the general procedure presented, among others, in [155]. FEM is based on the so-called weak formulation of the considered problem. To obtain the weak formulation, (2.14) is multiplied with test functions $\psi_j(\mathbf{r})$ and integrated over the problem domain Ω

$$\begin{aligned} &\int_\Omega x^{(k+1)}(\mathbf{r})\psi_j(\mathbf{r})\,\mathrm{d}\Omega - \int_\Omega x^{(k)}(\mathbf{r_c}(\mathbf{r}, t^{(k+1)}; t^{(k)}))\psi_j(\mathbf{r})\,\mathrm{d}\Omega \\ &+ D\Delta t \int_\Omega \nabla x^{(k+1)}(\mathbf{r}) \cdot \nabla\psi_j(\mathbf{r})\,\mathrm{d}\Omega = \Delta t \int_\Omega s^{(k+1)}(\mathbf{r})\psi_j(\mathbf{r})\,\mathrm{d}\Omega + \int_{\partial\Omega_N} D\psi_j(\mathbf{r})t_s(\mathbf{r})\,\mathrm{d}\Gamma. \end{aligned} \tag{2.15}$$

The equation can be solved using the standard finite element technique, in which the function $x(\mathbf{r})$ is approximated by the ansatz

$$x(\mathbf{r}) = \sum_{k=1}^{n} x_i\phi_i(\mathbf{r}), \tag{2.16}$$

with coefficients x_i to be determined and ansatz functions ϕ_i, where $i \in \{1,\ldots,n\}$. To obtain the vector of coefficients $\mathbf{x} = \mathrm{col}\{x_i\}_{i=1}^n$, the test functions $\psi_j(\mathbf{r})$ are replaced by the ansatz functions $\phi_j(\mathbf{r})$ according to the Galerkin approach. This yields a linear system for determination of the coefficients x_i:

$$(\mathbf{W_1} + \mathbf{W_2})\mathbf{x}^{(k+1)} - \mathbf{W_3}\mathbf{x}^{(k)} = \mathbf{f}^{(k+1)} \tag{2.17}$$

with

$$\mathbf{W_1} = \int_\Omega \phi_i\phi_j\,\mathrm{d}\Omega \tag{2.18}$$

$$\mathbf{W_2} = D\Delta t \int_\Omega \nabla\phi_i \cdot \nabla\phi_j\,\mathrm{d}\Omega \tag{2.19}$$

$$\mathbf{W_3} = \int_\Omega \phi_i(\mathbf{r_c}(\mathbf{r}, t^{(k+1)}; t^{(k)}))\phi_j\,\mathrm{d}\Omega \tag{2.20}$$

$$\mathbf{f} = \Delta t \int_\Omega s\phi_j\,\mathrm{d}\Omega + \Delta t D \int_{\partial\Omega_N} t_S\phi_j\,\mathrm{d}\Gamma. \tag{2.21}$$

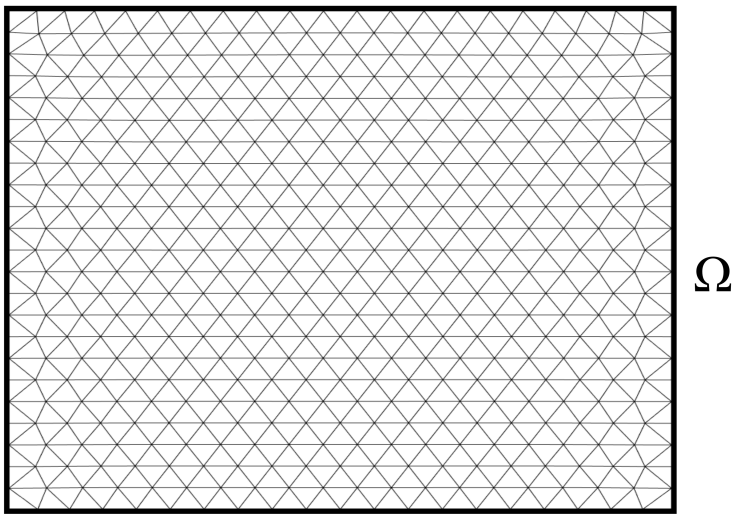


Figure 2.3: Exemplary FEM grid

Note that the Dirichlet conditions (2.7b) are not explicitly regarded in this formulation as they can be easily inserted into the complete system via boundary coefficients.

The ansatz functions ϕ_i are piecewise polynomial and result from a spatial discretization of the domain Ω into a set of non-overlapping cells called finite elements. An exemplary FEM grid is shown in Figure 2.3. In every element e, a polynomial ansatz function for the elemental solution $x^e(\mathbf{r})$ of the form

$$x^e(\mathbf{r}) = \sum_{j=1}^{n_e} x_j^e N_j^e(\mathbf{r}), \tag{2.22}$$

is defined, where the n_e local nodal variables x_j^e represent the function value (or in some cases also derivatives) at certain, prespecified positions $\mathbf{r}_j^e$ of the element, the so-called nodes. The local shape functions $N_j^e(\mathbf{r})$ have to fulfill the following condition at the nodes

$$N_j^e(\mathbf{r}_i) = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{if } j \neq i. \end{cases} \tag{2.23}$$

The choice of the local shape functions N_j^e depends on the type of element that is used.

In this work, triangular elements with six nodes ($n_e = 6$, see Figure 2.4) are considered. On the unit triangle, the local basisfunctions for this type of element yield

$$\begin{aligned} N_1^e &= \lambda_1(2\lambda_1 - 1), & N_4^e &= 4\lambda_1\lambda_2, \\ N_2^e &= \lambda_2(2\lambda_2 - 1), & N_5^e &= 4\lambda_2\lambda_3, \\ N_3^e &= \lambda_3(2\lambda_3 - 1), & N_6^e &= 4\lambda_3\lambda_1, \end{aligned} \tag{2.24}$$

where λ_i are the linear basis functions

$$\lambda_1 = 1 - \xi - \eta, \quad \lambda_2 = \xi, \quad \lambda_3 = \eta \tag{2.25}$$

with the local element coordinates ξ and $\eta \in [0, 1]$.

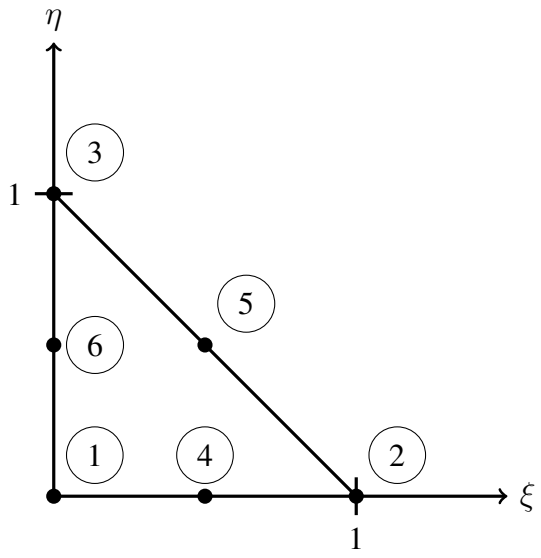


Figure 2.4: Quadratic triangular element with six degrees of freedom

This element is frequently applied in fluid flow problems and, thus, allows an easy transfer of potential additional fluid flow information to the considered pollutant dispersion problem. In general, the choice of the basis element is minor relevant for the methods and approaches that are developed in the following and can be easilyreplaced by other basis elementsis, thus, a good choice for the considered problems in this application as potential fluid information can be transferred easily.

Neighboring elements share nodes on their common edges so the global ansatz formulation (2.16) can be reconstructed by numbering all nodes consecutively so that the coefficients x_i become the nodal values. The global ansatz functions $\phi_i(\mathbf{r})$ are the composition of all local ansatz functions $N_j^e(\mathbf{r})$ evaluating to 1 at node $\mathbf{r}_i$. In general, it turns out that it is more practical to solve the resulting integrals locally on element level and to assemble the global system with the elemental integrals. Gaussian integration, which solves polynomials up to a certain degree exactly, can be used to solve the integrals in (2.18)-(2.21).

The integral in (2.20) needs special treatment since it depends on the characteristics. To consider the effect of the characteristics, it is examined, at which locations $\mathbf{r}_{S,i}^{(k)}$ the characteristics going through the nodes $\mathbf{r}_i$ at time $t^{(k+1)}$ intersect the time plane at $t^{(k)}$ (see Figure 2.5). An easy way to get the intersection points is to simply apply the Euler method

$$\mathbf{r}_{S,i}^{(k)} \approx \mathbf{r}_i - \Delta t \mathbf{w}(\mathbf{r}_i, t^{(k+1)}). \tag{2.26}$$

After determining in which element the intersection point $\mathbf{r}_{S,i}^{(k)}$ is located, the intersection point can be represented as a linear combination of the nodes $\mathbf{r}_{S_1}$, $\mathbf{r}_{S_2}$ and $\mathbf{r}_{S_3}$ of the triangle the intersection point is located in. Thus, it is possible to write $\mathbf{W}_3 = \mathbf{W}_1\mathbf{V}_2$ with a projection matrix $\mathbf{V}_2$ that contains the information of the characteristics.

To improve the performance of the algorithm, the expensive check in which triangle the intersection point is located is avoided within the scope of this work. Instead, the projection described above is only used if the intersection point $\mathbf{r}_{S,i}^{(k)}$ is located in a directly adjacent triangle of the node. If the intersection point does not lie in a neighboring triangle, the intersection point $\tilde{\mathbf{r}}_{S,i}^{(k)}$ of the characteristics with the space-time borders of the adjacent triangles are calculated (see Figure 2.6). In this case, the value at the

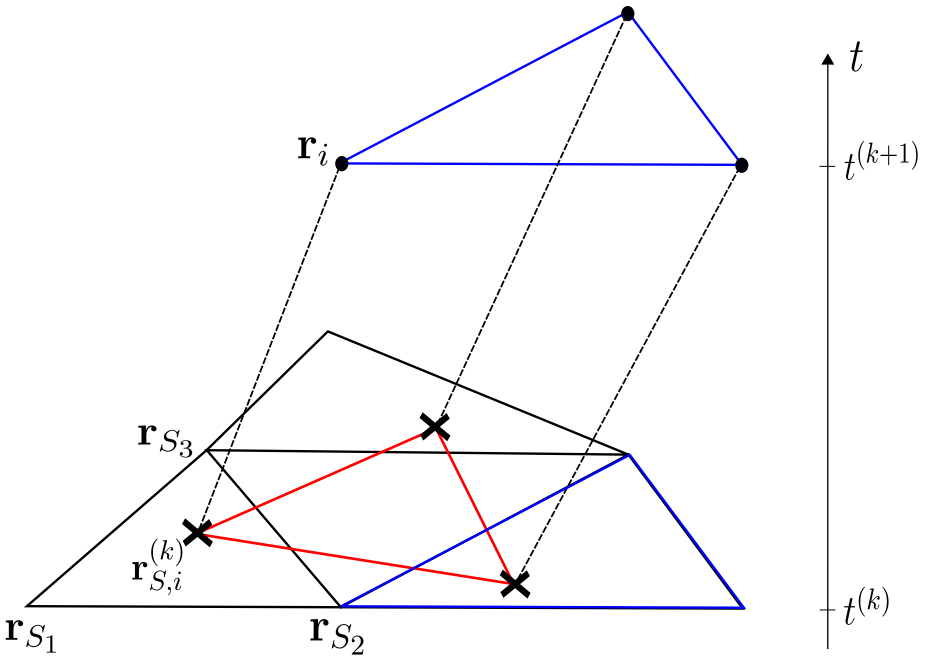


Figure 2.5: Illustration of the intersection point $\mathbf{r}_{S,i}^{(k)}$ between the characteristic going through $\mathbf{r}_i$ and the time plane at $t^{(k+1)}$

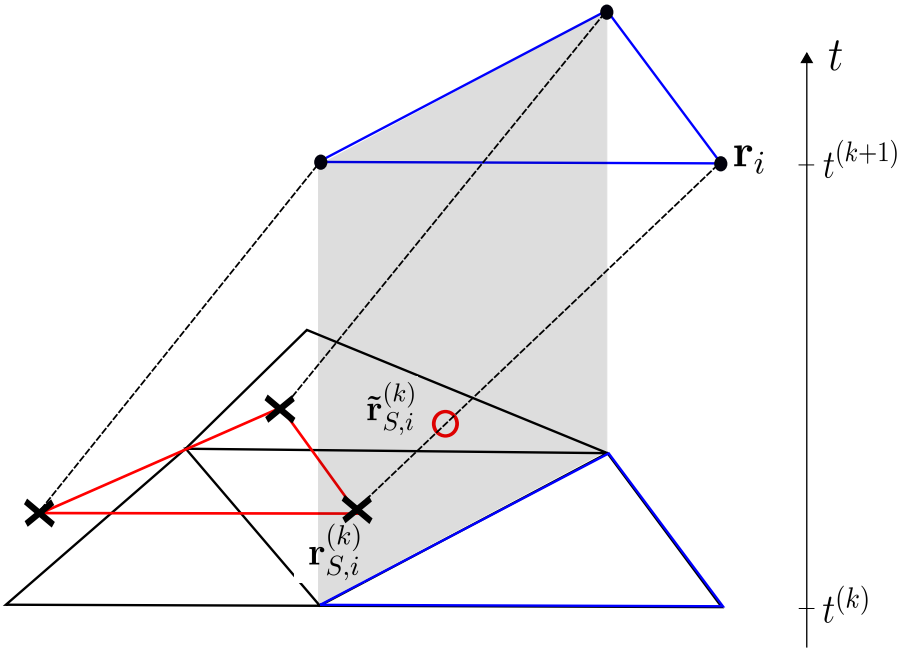


Figure 2.6: Illustration of the intersection point $\tilde{\mathbf{r}}_{S,i}^{(k)}$ between the characteristic going through $\mathbf{r}_i$ and the space-time plane

intersection point is represented by the nodal values of the old time stage $t^{(k)}$ and the new time stage $t^{(k+1)}$ so that the equation to be solved yields

$$(\mathbf{W}_1\mathbf{V}_1 + \mathbf{W}_2)\mathbf{x}^{(k+1)} - \mathbf{W}_1\mathbf{V}_2\mathbf{x}^{(k)} = \mathbf{f}^{(k+1)} \tag{2.27}$$

with another projection matrix $\mathbf{V}_1$. Given the results of the previous time stage $\mathbf{x}^{(k)}$, (2.27) represents a linear system to be solved for the new state vector $\mathbf{x}^{(k+1)}$. This can be achieved by a number of standard techniques.

2.3 Modeling of Sensor Vehicles

The model forecasts are complemented with measurements obtained from sensors mounted on vehicles that are able to adapt their motion subject to the current estimate. To design an operative monitoring approach, the dynamics of the sensor vehicles have to be taken into account. The vehicle dynamics considered in this work are described in Subsection 2.3.1. Furthermore, an observation model has to be defined. The used observation model is subject of Subsection 2.3.2 and a number of assumptions concerning observations are stated that facilitate the further observation treatment. Computation topology and communication aspects of the approach are discussed in Subsection 2.3.3.

2.3.1 Vehicle Dynamics

The motion dynamics of an unmanned vehicle applied in the monitoring scenario are generally described by a set of ordinary differential equations

$$\frac{\mathrm{d}\mathbf{p}(t)}{\mathrm{d}t} = \mathbf{f}(\mathbf{p}(t), \mathbf{w}(t), t), \tag{2.28}$$

where $\mathbf{p}(t)$ represents the state of the vehicle comprising its location, orientation, velocity, etc. The vehicle state $\mathbf{p}(t)$ can be influenced by the control input vector $\mathbf{w}(t)$. If the sensor vehicle should move to a specific location, the control input has to be chosen accordingly. To handle the vehicle dynamics computationally, (2.28) has to be discretized in time as presented in Section 2.6.

In general, the approaches presented within this work allow the application of every kind of sensor vehicle. The only restriction that has to be made is that the speed of the sensor vehicles should be, at least partially, larger than the typical process advection speed. In this way, it can be guaranteed that the sensor vehicles can reach informative measurement locations that are potentially advected with wind velocity. Typical vehicles that can be applied are for example fixed-wing aircrafts, quadrocopters or ground vehicles. It is even possible, to apply different kinds of vehicles in the proposed approaches. However, the aspect of heterogeneity as well as vehicles with more complex dynamics are not addressed in this work. The interested reader is referred to [54].

For simplicity, the dynamics of all sensor vehicles considered in this work are assumed to be governed by the two-dimensional double-integrator dynamics

$$\frac{\mathrm{d}\mathbf{p}(t)}{\mathrm{d}t} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \mathbf{p}(t) + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{u}(t), \tag{2.29}$$

where the sensor state vector $\mathbf{p}$ comprises the current sensor position $\mathbf{r^p} = (r_1^p, r_2^p)^T$ as well as the current sensor velocity in r_1- and r_2-direction $\mathbf{v^p} = (v_1^p, v_2^p)^T$. The control input $\mathbf{u}$ consists of the acceleration in both directions. These dynamics present an approximation to the dynamics of quadrocopters and provide a good trade-off between a simple and focused development of the monitoring approaches and complexity of the dynamics. Still, it is straightforward to apply more complex vehicle dynamics for the approaches presented in this work – the linearized vehicle model of the vehicle controller (see Subsection 2.6) is the only part to be modified.

2.3.2 Observation Model

In general, the forecasted state vector $\mathbf{x}^{(k)}$ introduced in Section 2.2 does not correspond to the true concentration state $\mathbf{x^t}^{(k)}$, i.e. to the actual concentration values present at the position of the nodes. However, the true state can be accessed via the measurements provided by the sensors. At discrete times $t^{(k)}$, a sensor takes a measurement $y^{(k)}$ of the current concentration value at the location it is currently positioned. This relation can be expressed by the observation model

$$y^{(k)} = h^{(k)}(\mathbf{x^t}^{(k)}) + \epsilon^{(k)}. \tag{2.30}$$

According to this model, the observation $y^{(k)}$ depends on the observation model operator $h^{(k)}$ applied to the true concentration state $\mathbf{x^t}^{(k)}$. The observation model operator $h^{(k)}$ is dependent on the current sensor position $\mathbf{r^p}^{(k)}$ and outputs the part of the true state that becomes available to the sensor. Due to instrumental and representativeness errors, an observation error $\epsilon^{(k)}$ cannot be avoided and is added to every measurement.

Typically, more than one measurement is available so that (2.30) can be extended to the multi-dimensional case with n_p measurements

$$\mathbf{y}^{(k)} = \mathbf{h}^{(k)}(\mathbf{x^t}^{(k)}) + \boldsymbol{\epsilon}^{(k)}. \tag{2.31}$$

with $\mathbf{y}^{(k)} = \text{col}\{y_j^{(k)}\}_{j=1}^{n_p}$, $\boldsymbol{\epsilon}^{(k)} = \text{col}\{\epsilon_j^{(k)}\}_{j=1}^{n_p}$ and $\mathbf{h}^{(k)}(\mathbf{x^t}^{(k)}) = \text{col}\{h_j^{(k)}(\mathbf{x^t}^{(k)})\}_{j=1}^{n_p}$.
For the remainder of this work, a number of assumptions are made that simplify the treatment of the observations

- *Concentration is measured directly:* If the concentration value, i.e. the entity we are looking for, is measured directly, the observation model operator $\mathbf{h}$ becomes linear and the $n_p \times n$-dimensional matrix $\mathbf{H}$ can be used instead

  $$\mathbf{y}^{(k)} = \mathbf{H}^{(k)}\mathbf{x^t}^{(k)} + \boldsymbol{\epsilon}^{(k)}. \tag{2.32}$$

  The matrix $\mathbf{H}$ contains only a few non-zero entries, namely at locations near current sensor positions.
- *Gaussian, unbiased observation error:* The observation error ϵ is assumed to be Gaussian with zero mean and a known error variance R. Prior to the monitoring approach, the quality of the measurement devices can be assessed and the measurement parameters can be calibrated so that this assumption is not severe.
- *Observation errors are uncorrelated:* If observation errors are uncorrelated, the observation error covariance matrix $\mathbf{R}$ corresponding to the observation error $\boldsymbol{\epsilon}$ is diagonal simplifying further calculations. Typically, the measurements of different sensor vehicles are not related so that this assumption is also justified.

- *Measurements at same time:* For simplicity, it is assumed in the following that each sensor vehicle produces a measurement at the same time. If this is not the case and measurements become asynchronous, some modifications have to be made dating measurements back to previous times. These modifications lie outside the scope of this work.

2.3.3 Sensor Network and Communication

The presented approach is centralized. This means that the lion's share of all the calculations that have to be performed is executed on a central supercomputer. Simulation, data assimilation, target generation and vehicle control are all tasks performed centrally. As the on-board computational power is limited and a central computer is equipped with high computational power and memory, this work distribution seems to be suitable. However, this also means that the information available at the sensor vehicles, e.g. concentration measurements and vehicle state information, has to be transferred to the central station. Moreover, the control inputs that are calculated at the central point must be sent back to the sensor vehicles. This results in the star topology network depicted in Figure 2.7. Communication between central station and sensor vehicles has to be maintained repeatedly and in every step.

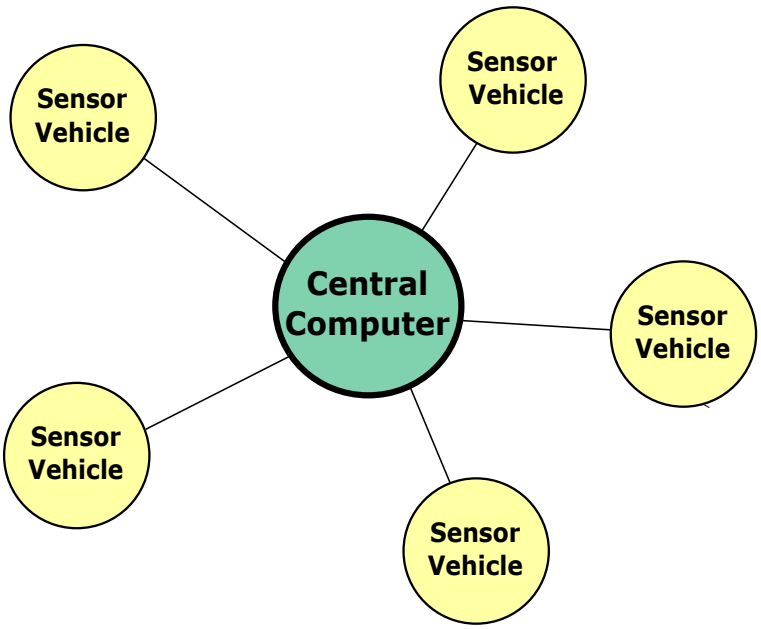


Figure 2.7: Communication network of the proposed monitoring approach

Usually communication standards as IEEE 802.15 or wifi (IEEE 802.11b/g) are used in wireless sensor networks [3, 197]. While bandwidth is, at least for a moderate number of sensor vehicles, not a critical issue since only lightweight information are exchanged, range definitely is. Typical communication ranges are between 100 and 200 meters. In scenarios with a length scale of several kilometers, the base station is likely to be out of reach. In this cases, additional intermediate devices have to be placed or, alternatively, multi-hop has to be enabled. The latter case would require a change in sensor topology and results in an additional constraint for vehicle motion, which is not considered in the scope of this work. Instead, perfect communication without loss is assumed in this chapter.

2.4 Data Assimilation

An important step in the monitoring process is the fusion of measurements and model information. Integrating the observations of the sensors into the current model reduces uncertainty and increases the quality of the considered estimates. In this way, the problem of unknown initial conditions, imprecise process parameters, and model and observation errors can be compensated enabling much more precise forecasts. The question of how to best merge the information obtained from the model on the one side and from the observations on the other side is answered by the concept of data assimilation described in this section. After a more detailed description of the data assimilation problem and a review of possible methods to tackle it in Section 2.4.1, the Kalman Filter, one of the most popular data assimilation methods in practice is presented in Section 2.4.2. As the use of the Kalman Filter is costly in scenarios with high dimensional state vectors, ensemble Kalman methods seem to be more suitable. For this reason, the modifications to the Kalman Filter that are necessary to employ the Ensemble Kalman Filter (EnKF) and the Ensemble Square Root Filter (EnSRF) are presented in Section 2.4.3. Furthermore, aspects important in context of handling ensembles are described – in particular, a method to treat state-dependent model error is proposed. The strategy to combine state and parameter estimation yielding a joint state and parameter estimation process is subject of Section 2.4.4 and is applied to the problem of source estimation in the considered application.

2.4.1 Problem Description and Methodology

The goal of data assimilation is to estimate the current true process state $\mathbf{x}^{\mathbf{t}(k)}$ based on the available observations $\mathbf{y}^{(k)}$ and a state forecast $\mathbf{x}^{\mathbf{f}(k)}$ with the superscript $(\cdot)^f$ indicating a forecasted entity. In this work, the forecast is obtained using the model forecast equation (2.27) on the prior estimate $\mathbf{x}^{(k-1)}$, i.e.

$$\mathbf{x}^{\mathbf{f}(k)} = \mathbf{M}\mathbf{x}^{(k-1)} + \mathbf{f}^{(k)} \tag{2.33}$$

with $\mathbf{M} = (\mathbf{W}_1\mathbf{V}_1 + \mathbf{W}_2)^{-1}\mathbf{W}_1\mathbf{V}_2$.

Both measurements and forecasts are afflicted with errors and the motivation is to obtain an estimate with less error after the assimilation procedure. In general, the following sources of errors occur:

- *Observation errors:* When taking measurements, instrumental and representativeness errors result in an observation error $\boldsymbol{\epsilon}^{(k)}$ as already indicated in (2.31). It must be noted that the resulting measurement vector does not correspond to the actual real world values at the observation sites, but deviates from them according to the observation error. Thus, the concentration value at the observation sites should not be simply set to the measured value, but also the information of a background state (i.e. the forecasted state) as well as statistics of the corresponding observation error should be considered in the assimilation procedure.
- *Model errors:* Naturally, the model is not able to consider all physical effects and suffers from unexpected external perturbations so that the model forecast does not correspond to the true forecast

  $$\mathbf{x}^{\mathbf{t}(k)} = \mathbf{M}\mathbf{x}^{\mathbf{t}(k-1)} + \mathbf{f}^{(k)} + \boldsymbol{\eta}^{(k)}. \tag{2.34}$$

  This means that a model error $\boldsymbol{\eta}^{(k)}$ is made in every prediction step leading to an increasing deviation between the forecast and the true state from step to step if no measurements are integrated.

Therefore, it is important to complement the model forecast with measurements. Besides unmodelled physics and external perturbations, the model error also contains numerical errors as well as errors in boundary conditions and process parameters.

- *Errors in boundary conditions:* A special part of the model error are errors in the boundary conditions. A slightly wrong choice of, for example, the type of boundary condition or the concentration at the inlet will also lead to a deviation between true and forecasted state. For the remainder of this work, errors in boundary conditions are not considered separately and are treated as a part of the model error. However, it should be paid attention to the fact that the model error at the boundaries should be assumed to be high if boundary conditions are uncertain.
- *Errors in parameters:* Another part of the model error are errors in the process parameters. In most scenarios, process parameters like the wind velocity or the diffusion coefficient are not exactly known and may differ temporally and spatially from the value that is assumed. This also leads to an increasing error in the forecast. While the parameter error is also treated to be included in the model error for some parts of this work, it is explicitly considered when estimating parameters along with the process state (see Subsection 2.4.4).
- *Errors in initial conditions:* Errors in initial conditions are not part of the model error, but determine the error at the begin of the simulation. In most scenarios, only a very rough estimate of the initial state is known so that the error in the initial conditions is rather high. Without the use of measurement information, this error will most likely increase over time, even if the model error is zero.

In order to alleviate the influence of the errors presented above, the key idea of data assimilation is the combination of observation and model forecast. This process is called analysis and the resulting estimate is the so-called analysis state vector $\mathbf{x}^{\mathbf{a}(k)}$ with the superscript $(\cdot)^a$ representing entities after the analysis.

A number of different methods to perform the analysis have been proposed starting in the 18th century when Carl Friedrich Gauss presented the least squares method [110]. Another early approach is the Cressman analysis where the analysis state is set to the observation in the vicinity of the observation locations and to the forecasted state far away from them [21]. While this method is rather simple, it has a decisive drawback: Accurate prior estimates can be replaced by poor quality observations. For this reason, the use of statistical methods considering the uncertainty of the available data is highly recommendable.

A great breakthrough in this context has been the development of the Kalman Filter [95, 96] by Rudolf Kalman in the 1960s. Originally developed for the spacecraft industry, the Kalman Filter, together with its numerous modifications, is still today one of the most frequently used assimilation methods in navigation, control of vehicles and signal processing. Based on the assumption that observation error and model error are unbiased and normally distributed with known error covariance matrices, the state forecast is linearly corrected by the weighted innovation of the observations. The weight that is used in this approach relies on the error covariance matrix of the mean estimate that is also forecasted and modified in the analysis phase.

The Kalman Filter is the basis of a number of extensions that have been proposed, especially by the numerical weather prediction community dealing with high problem dimensions. If the state vector is high-dimensional, the computation of the error covariance matrix might become very costly. Ensemble methods like the Ensemble Kalman Filter or Ensemble Square Root Filters [57, 178, 79] do not consider the propagation of the full covariance matrix but represent the covariance matrix implicitly by a set of perturbed state vectors. Another popular type of methods for high dimensional models are variational

approaches, such as 3D-Var and 4D-Var [113, 48]. Variational methods are vector-based data assimilation techniques relying on the solution of a variational optimization problem minimizing the sum of the distances to the forecast and the observation respectively weighted by an error covariance matrix. A major drawback of these methods is the fact that it is not intuitive to construct the error covariance matrices to be applied and that the error covariance matrices are not propagated in time – a characteristic that is needed for the dynamic data-driven monitoring approach as shown later.

The use of zero-mean Gaussian error distributions, especially for the model error, is sometimes problematic since the actual error has another probability density function. Model drift and bad quality estimates can result. Methods like the Particle Filter [70, 50] allow for non-Gaussian errors as they approximate the distribution by a large number of samples, but they are computationally much too expensive for the scenarios considered in this work.

In this work, Kalman and ensemble Kalman methods are applied. They provide a means to maintain an error covariance matrix that allows estimating the current uncertainty of the system, an important measure for data-driven monitoring. For reasons of computational tractability, especially ensemble methods seem to be suitable for high-dimensional systems.

Before taking a closer look at the different data assimilation schemes, the model and observation error assumptions considered for most parts of this work are summarized

- *Gaussian, unbiased model error:* The model error $\boldsymbol{\eta}$ is Gaussian with zero mean and possesses a known error covariance matrix $\mathbf{Q}$. This assumption is very critical since the actual model errors usually deviate from a normal distribution, especially if the concentration, which cannot become negative, is the considered entity. However, the use of the Gaussian assumption represents a compromise between accuracy and efficient computability. If it is absolutely necessary to take the non-negativity of the concentration into account, the gamma or inverse gamma distribution as well as the log-normal distribution could be suitable probability density functions. For this kind of distributions, extensions of the filters used in the following have been recently proposed [201, 18], but are outside the scope of this work.

  In general, the model error covariance matrix $\mathbf{Q}$ is not known, but, as a rule of thumb, one can use that correlation between variables decays with increasing distance between them. Furthermore, it will be shown in the following that it is beneficial in the considered applications to choose a state-dependent model error covariance matrix.

  For the later test case scenarios, the model error occuring in the propagation of the true process state will not be set to a normally distribution to provide more realistic conditions and to test how good the applied methods based on this assumptions are performing.
- *Gaussian, unbiased, uncorrelated observation error:* Referring to the assumptions of Subsection 2.3.2, the observation error $\boldsymbol{\epsilon}$ is also Gaussian with zero mean and known error covariance matrix $\mathbf{R}$, which is diagonal since the observation errors are uncorrelated.
- *Observation errors and model errors are uncorrelated:* The errors made in the measurement process are typically independent of the errors of the forecast method so that this assumption, needed for the derivation of the Kalman Filter, is usually fulfilled.

2.4.2 Kalman Filter

The Kalman Filter is probably one of the most popular data assimilation techniques [192]. Developed in the 1960s [95, 96], it is applied in many fields, e.g. in robotics, signal processing, and aerospace

engineering. The Kalman Filter is a sequential and recursive assimilation technique. New observations are processed immediately and the updates for the new analysis are solely based on the prior estimate, not on the whole history of the estimate. As a result of the Kalman Filter method, one obtains an estimate of the mean state $\mathbf{x}^{\mathbf{a}}$ of the considered process as well as the estimate's error covariance matrix $\mathbf{P}^{\mathbf{a}}$.

The forecast operator $\mathbf{m}$ as well as observation operator $\mathbf{h}$ are required to be linear and the observation error $\boldsymbol{\epsilon}$ and the model error $\boldsymbol{\eta}$ must be uncorrelated, Gaussian, and unbiased. As described above, all these requirements hold for the considered case so that the Kalman Filter from the assumption's point of view can be applied in the considered scenarios. By the way, an extension to a nonlinear model case is not bristled with difficulties. In that case, the nonlinear form of the Kalman Filter, the so-called Extended Kalman Filter [88, 66, 92], dealing with linearizations of the model operator in every step, or an ensemble variant (cf. Section 2.4.3) is an adequate substitute.

The Kalman Filter consists for every analysis cycle of a prediction and an update step. In the prediction step, the mean state estimate and its error covariance matrix are propagated to the time $t^{(k)}$, in which new observations are available. To achieve this, the forecast operator is applied to the mean analysis state from the previous observation time using (2.33). Please note that it is assumed in the following that only one simulation step is necessary to step from one observation time to the next. If observation rate is slower, one can also perform several simulation steps to forecast the state estimate to the next observation time. Besides the mean estimate, the error covariance has to be propagated to the current time $t^{(k)}$ using the equation

$$\mathbf{P}^{\mathbf{f}(k)} = \mathbf{M}\mathbf{P}^{\mathbf{a}(k-1)}\mathbf{M}^T + \mathbf{Q}^{(k)}. \tag{2.35}$$

The model error covariance matrix $\mathbf{Q}^{(k)}$ has to be added in every step.

The second step is the update step that corrects the forecast estimate and the error covariance matrix by incorporating the information obtained from the observations (2.32). In the Kalman Filter, the forecast $\mathbf{x}^{\mathbf{f}(k)}$ is linearly combined with the innovation vector $\mathbf{d}^{(k)} = \mathbf{y}^{(k)} - \mathbf{H}^{(k)}\mathbf{x}^{\mathbf{f}(k)}$ weighted by the matrix $\mathbf{K}^{(k)}$ so that the analysis reads

$$\mathbf{x}^{\mathbf{a}(k)} = \mathbf{x}^{\mathbf{f}(k)} + \mathbf{K}^{(k)}\mathbf{d}^{(k)}. \tag{2.36}$$

Accordingly, the observation has an effect on the error covariance matrix. Thus, the analysis error covariance matrix has the form

$$\mathbf{P}^{\mathbf{a}(k)} = (\mathbb{1} - \mathbf{K}^{(k)}\mathbf{H}^{(k)})\mathbf{P}^{\mathbf{f}(k)} \tag{2.37}$$

with the $n \times n$-dimensional identity matrix $\mathbb{1}$. The choice of the Kalman gain $\mathbf{K}^{(k)}$ is crucial since it represents the influence of the innovation on the analysis and steers the extent to which observational information is included in the analysis. The basic idea of the Kalman Filter is to choose $\mathbf{K}^{(k)}$ in such a way that the analysis error covariance (2.37) is minimized. It turns out (see [110] for a derivation) that the optimal Kalman gain is

$$\mathbf{K}^{(k)} = \mathbf{P}^{\mathbf{f}(k)} \left(\mathbf{H}^{(k)}\right)^T \left(\mathbf{H}^{(k)}\mathbf{P}^{\mathbf{f}(k)} \left(\mathbf{H}^{(k)}\right)^T + \mathbf{R}^{(k)}\right)^{-1}. \tag{2.38}$$

All things considered, the update scheme of the Kalman Filter comprises the calculation of the Kalman gain (2.38) and the update of the mean state estimate (2.36) as well as the error covariance matrix (2.37) based on the Kalman gain. The resulting analysis state and covariance matrix build the basis for the next

cycle, i.e. mean and covariance are propagated to the next observation time and updated with the new observations and so on.

Maintaining the error covariance matrix is a difficult issue when dealing with high-dimensional models. Considering memory aspects, an $n \times n$-dimensional symmetric matrix has to be stored and regarding computation time, especially the propagation (2.35) can become problematic involving the multiplication of high-dimensional, dense matrices. Therefore, methods avoiding the explicit computation of the error covariance matrix should be applied in scenarios with a high problem dimension.

2.4.3 Ensemble Methods

One class of approaches that avoids the explicit maintenance of the error covariance matrix are ensemble methods. Ensemble methods represent the error covariance matrix implicitly by propagating a set of perturbed state vectors. In the following, two Ensemble methods are presented: The Ensemble Kalman Filter that uses a stochastic update approach and, based thereupon, the Ensemble Square Root Filter that provides a deterministic ensemble update step.

Ensemble Kalman Filter

The Ensemble Kalman Filter (EnKF) was developed by Evensen in 1994 [57] and is a sequential data assimilation scheme. It extends the Kalman Filter by a Monte Carlo method so that it is also capable of handling nonlinear problems without linearization. The main advantage of applying the EnKF in the context of this work is that the EnKF is much more efficient and much less expensive than the original Kalman Filter if high dimensional problems are considered. The reason for this efficiency is based upon the fact that the greedy covariance propagation of the Kalman Filter is avoided. Instead, an ensemble of state estimates, i.e. a set of $n_s \ll n$ perturbed state vectors $\{\mathbf{x}_{(i)}\}_{i=1}^{n_s}$, is used to represent the mean estimate as well as the error covariance matrix implicitly.

While the mean state can be expressed by the ensemble as

$$\bar{\mathbf{x}} = \frac{\sum_{i=1}^{n_s} \mathbf{x}_{(i)}}{n_s}, \tag{2.39}$$

the error covariance matrix is represented by the deviations from the ensemble mean

$$\mathbf{P} = \frac{\sum_{i=1}^{n_s} (\mathbf{x}_{(i)} - \bar{\mathbf{x}})(\mathbf{x}_{(i)} - \bar{\mathbf{x}})^T}{n_s - 1}. \tag{2.40}$$

Like the original KF, the EnKF consists of prediction and correction steps with a prediction step similar to the prediction step of the Kalman Filter. The difference is that the forecast equation (2.33) is not only used for a single state forecast, but for the forecast of each ensemble member $i \in \{1, \ldots, n_s\}$

$$\mathbf{x}_{(i)}^{\mathbf{f}(k)} = \mathbf{M}\mathbf{x}_{(i)}^{\mathbf{a}(k-1)} + \mathbf{f}^{(k)}. \tag{2.41}$$

As mentioned before, the propagation of the covariance matrix (2.35) can be skipped. Instead, the forecast error covariance matrix $\mathbf{P}^{\mathbf{f}}$ can be computed using (2.40).

The forecast error covariance matrix can then be used in the update step to calculate the Kalman Gain $\mathbf{K}$ following (2.38). With the Kalman gain, all ensemble members can be updated individually according to (2.36)

$$\mathbf{x}_{(i)}^{\mathbf{a}(k)} = \mathbf{x}_{(i)}^{\mathbf{f}(k)} + \mathbf{K}^{(k)} \left(\mathbf{y}^{(k)} - \mathbf{H}^{(k)} \mathbf{x}_{(i)}^{\mathbf{f}(k)} \right). \tag{2.42}$$

Again, the analysis error covariance matrix can be determined by (2.40) if needed.

Although originally proposed in the form above, it is recommended to use a slightly modified version of the EnKF update (2.42). To prevent the ensemble from degeneration, every ensemble member should not be updated with the actual observation vector $\mathbf{y}^{(k)}$, but with a perturbed observation vector $\tilde{\mathbf{y}}^{(k)}$ adding noise randomly sampled from the observation error distribution. In this way different ensemble members are updated using different perturbed observations and the effect of the observation error on the ensemble is taken into account [27, 80]. For this reason, the EnKF is classified to be a stochastic filter.

To get an update of the error covariance matrix, the actual measurements have to be known. The issues regarded in this work, however, require the ability to assess the effects of potentially unknown measurements on the error covariance matrix. For this reason, further extensions of the EnKF should be considered and deterministic filters [7, 19, 193, 178] should be taken into account.

Ensemble Square Root Filter

The Ensemble Square Root Filter [193, 178] is a deterministic filter that avoids the perturbation of the observation vector and is able to update all ensemble members in one step.

While the forecast step is identical to the forecast step of the EnKF and (2.41) is used to propagate the ensemble members to observation time, the analysis step is different. Defining the $n \times n_s$-dimensional deviation matrix

$$\mathbf{X} = \text{row} \left\{ \frac{\mathbf{x}_{(i)} - \bar{\mathbf{x}}}{\sqrt{n_s - 1}} \right\}_{i=1}^{n_s}, \tag{2.43}$$

the forecast error covariance matrix can be written as

$$\mathbf{P}^{\mathbf{f}(k)} = \mathbf{X}^{\mathbf{f}(k)} \mathbf{X}^{\mathbf{f}(k)^T}, \tag{2.44}$$

i.e. $\mathbf{X}^{\mathbf{f}}$ is a matrix square root of $\mathbf{P}^{\mathbf{f}}$.

The first step of the analysis consists of the update of the ensemble mean $\bar{\mathbf{x}}^{\mathbf{f}}$ to the updated ensemble mean $\bar{\mathbf{x}}^{\mathbf{a}}$. The standard correction formulas (2.38) and (2.36) can be applied with a simplification in the computation of the Kalman gain (2.38): The error covariance matrix does not have to be determined explicitly, but applying the observation model matrix $\mathbf{H}$ to the perturbation matrix $\mathbf{X}$ before multiplying the perturbation matrices simplifies the calculation process

$$\mathbf{K}^{(k)} = \mathbf{X}^{\mathbf{f}(k)} (\mathbf{H}^{(k)} \mathbf{X}^{\mathbf{f}(k)})^T \left(\mathbf{H}^{(k)} \mathbf{X}^{\mathbf{f}(k)} (\mathbf{H}^{(k)} \mathbf{X}^{\mathbf{f}(k)})^T + \mathbf{R}^{(k)} \right)^{-1}. \tag{2.45}$$

As (2.44) should also hold after the analysis, i.e. $\mathbf{P}^{\mathbf{a}(k)} = \mathbf{X}^{\mathbf{a}(k)} (\mathbf{X}^{\mathbf{a}(k)})^T$, the analysis ensemble has to be determined adequately. Inserting (2.44) in (2.37) yields

$$\mathbf{X}^{\mathbf{a}(k)} = \mathbf{X}^{\mathbf{f}(k)} \mathbf{T}^{(k)} \tag{2.46}$$

with a transformation matrix $\mathbf{T}^{(k)}$ fulfilling the relation

$$\mathbf{T}^{(k)} \left(\mathbf{T}^{(k)}\right)^T = (\mathbb{1} - \left(\mathbf{H}^{(k)}\mathbf{X}^{\mathbf{f}(k)}\right)^T \left(\mathbf{D}^{(k)}\right)^{-1} \mathbf{H}^{(k)}\mathbf{X}^{\mathbf{f}(k)}) \tag{2.47}$$

with $\mathbf{D}^{(k)} = \mathbf{H}^{(k)}\mathbf{X}^{\mathbf{f}(k)} \left(\mathbf{H}^{(k)}\mathbf{X}^{\mathbf{f}(k)}\right)^T + \mathbf{R}^{(k)}$. This means that the transformation matrix $\mathbf{T}^{(k)}$ is obtained by calculating the matrix square root of the $n_s \times n_s$-dimensional matrix $\mathbb{1} - \left(\mathbf{H}^{(k)}\mathbf{X}^{\mathbf{f}(k)}\right)^T \left(\mathbf{D}^{(k)}\right)^{-1} \mathbf{H}^{(k)}\mathbf{X}^{\mathbf{f}(k)}$. Also, the calculation of $\left(\mathbf{D}^{(k)}\right)^{-1}$ is not very costly as the number of measurements n_p is typically small.

Having computed the analysis perturbation matrix, the analysis ensemble set can be reconstructed

$$\mathbf{x}_{(i)}^{\mathbf{a}(k)} = \bar{\mathbf{x}}^{\mathbf{a}(k)} + \left(\mathbf{X}^{\mathbf{a}(k)}\right)_{\cdot,i}. \tag{2.48}$$

Using the EnSRF does not require the actual measurements to update the error covariance matrix and is, therefore, perfectly applicable in AO scenarios. Furthermore, the deterministic update method avoids sampling errors associated with perturbing observations and is considered more efficient and less expensive than the pure Ensemble Kalman Filter since the full error covariance matrix does not have to be computed at any place.

Practical Work with Ensemble Filters

Working with Ensemble Filters means approximating the full error covariance matrix $\mathbf{P}$ by a lower rank matrix constructed from the deviations of an ensemble from its mean. To perform these approximations, some parameters or methodologies have to be chosen to construct the ensemble on the one hand and to retain a representative ensemble during the simulation on the other hand.

Before applying an Ensemble Filter, the size n_s of the ensemble has to be determined. If n_s is small, computational time will be saved, but the representativeness of the error covariance matrix will be decreased. If n_s is large, the ensemble will be more representative at the cost of a higher computation time.

In general, the optimal ensemble size certainly depends first of all on the problem dimension and the scenario. However, a typical ensemble size amounts to 100 or less in a number of very high-dimensional simulations. If a threshold ensemble size is reached, the EnKF works successfully in most cases [68].

After having chosen the size of the ensemble, the ensemble members have to be generated. A generic way is to just sample the ensemble members from the initial estimate's statistics, i.e. from the initial mean state vector and from the initial error covariance matrix. However, when n_s is small, the future application might suffer from sampling errors. For this reason, an eigendecomposition of the initial error covariance matrix $\mathbf{P}^{(0)} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ is applied, using the n_s dominating eigenvectors $\mathbf{v}_i$ for the initial ensemble [137, 59]:

$$\mathbf{x}_{(i)}^{(0)} = \bar{\mathbf{x}}^{(0)} + \sqrt{(n_s - 1)\lambda_i}\mathbf{v}_i. \tag{2.49}$$

To guarantee that $\bar{\mathbf{x}} = \mathbf{x}^{(0)}$, the difference between ensemble mean and initial mean estimate is subtracted from every ensemble member.

Other problems might occur during the simulation caused by a small ensemble size. Error covariance matrices constructed by a small number of ensembles members often contain spurious correlations between

states that are far away from each other [80]. Unphysical updates caused by measurements far away from the concerning location can occur. A strategy to tackle this problem is called localization [73, 81]. Localization methods restrict the influence of an observation to its vicinity by weighting the covariance matrix with a distance function with local compact support [63]. Using the $n \times n_p$-dimensional localization matrix $\mathbf{\Psi}_1$ and the $n_p \times n_p$-dimensional localization matrix $\mathbf{\Psi}_2$, both matrices being based on the distance functions, the calculation of the Kalman gain (2.38) can be replaced by

$$\mathbf{K}^{(k)} = \mathbf{\Psi_1} \circ (\mathbf{P}^{\mathbf{f}(k)} (\mathbf{H}(k))^T) \left(\mathbf{\Psi_2} \circ \left(\mathbf{H}^{(k)} \mathbf{P}^{\mathbf{f}(k)} \left(\mathbf{H}^{(k)} \right)^T \right) + \mathbf{R}^{(k)} \right)^{-1} \tag{2.50}$$

with the element-wise Schur product $(\mathbf{A} \circ \mathbf{B})_{i,j} = (\mathbf{A})_{i,j} (\mathbf{B})_{i,j}$. Another possibility to avoid long-range correlations is to assimilate data in local physical space [58, 132, 83]. Only measurements corresponding to locations in the vicinity are used to calculate the update at every location.

To this point, the model error $\boldsymbol{\eta}$ has not been treated in the formulation of the ensemble filters so that the filters described above underestimate the true forecast error. This might lead to filter divergence since it is much more trusted in the forecast than in new measurements. One method to remedy this problem is to simply add random perturbation vectors drawn from the model error distribution to every ensemble member. However, the generation of random vectors from the model error distribution requires a lot of computation time when dealing with high-dimensional models. Another option is to apply multiplicative inflation [8]. To prevent the ensemble from becoming degenerated, the ensemble deviations are multiplied by a scalar α, which is slightly larger than one

$$\tilde{\mathbf{X}}^{\mathbf{f}} = \alpha \mathbf{X}^{\mathbf{f}}. \tag{2.51}$$

In this way, the ensemble members are pushed away from each other and a sufficiently representative ensemble set can be maintained. However, such a model of the model error might not adequately describe the real behavior of the model error. It is likely that the model error in regions with a high concentration is higher than in regions, in which the concentration is low. Thus, it is proposed that the model error can be approximated by the relation

$$\mathbf{Q}^{(k)} = \mathbf{\Psi} \circ \left(\bar{\mathbf{z}}^{(k)} \left(\bar{\mathbf{z}}^{(k)} \right)^T \right) \tag{2.52}$$

with $\bar{\mathbf{z}}^{(k)} = \text{col}\{\text{sgn}(\bar{\mathbf{x}}^{(k)})_i \sqrt{|(\bar{\mathbf{x}}^{(k)})_i|}\}_{i=1}^{n}$ and a constant distance-depending matrix $\mathbf{\Psi}$ similar to the localization matrices $\mathbf{\Psi_1}$ and $\mathbf{\Psi_2}$. Instead of computing $\mathbf{Q}^{(k)}$ in every step and performing a computationally demanding draw from the associated probability distribution, the n_s most dominant eigenvectors $\mathbf{v}_i$ of $\mathbf{\Psi}$ are computed prior to the application and weighted by the square root of the respective eigenvalue λ_i and the ensemble size n_s

$$\tilde{\mathbf{q}}_i = \sqrt{(n_s - 1)\lambda_i} \mathbf{v}_i. \tag{2.53}$$

With the zero-mean vectors $\mathbf{q}_i = \tilde{\mathbf{q}}_i - \bar{\tilde{\mathbf{q}}}$ element-wisely multiplied with the current square root mean state $\bar{\mathbf{z}}^{(\mathbf{k})}$, one can construct the $n \times n_s$-dimensional matrix

$$\mathbf{\Upsilon}^{(k)} = \text{row}\{\bar{\mathbf{z}}^{(k)} \circ \mathbf{q}_i\}_{i=1}^{n_s}, \tag{2.54}$$

with the matrix product $\mathbf{\Upsilon}^{(k)} \left(\mathbf{\Upsilon}^{(k)} \right)^T$ approximating the model error covariance matrix $\mathbf{Q}^{(k)}$. Thus, in every forecast step, the model error is taken into account by adding the respective column $\left(\mathbf{\Upsilon}^{(k)} \right)_{\cdot,i}$

to every ensemble member $\mathbf{x}_{(i)}^{(k)}$. Please note that the addition with $\boldsymbol{\Upsilon}^{(k)}$ does not change the mean of the ensemble set. Furthermore, it should be noted that even if $\mathbf{X}\mathbf{X}^T = \mathbf{P}$ and $\boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^T = (n_s - 1)\mathbf{Q}$, in general $(\mathbf{X} + \boldsymbol{\Upsilon})(\mathbf{X} + \boldsymbol{\Upsilon})^T \neq \mathbf{P} + (n_s - 1)\mathbf{Q}$. However, since the actual values of $\mathbf{Q}^{(k)}$ are not exactly known anyway, the proposed approach is considered to be a good approximation for a state-dependent model error.

2.4.4 Joint State-Parameter Estimation

Not only the process state is unknown in the considered scenarios, also a number of process parameters are not known exactly. The poor knowledge of source function, background velocity, diffusivity, etc. represents another source of uncertainty. Moreover, important process parameters are required for a more detailed assessment of the current situation. Thus, parameters should be estimated together with the state vector [60].

Parameter estimation is usually separated from state estimation. For instance, a number of parameters are estimated off-line prior to the actual simulation using model calibration. However, important process parameters like the source function, for example, can not be calibrated in this way and an online methodology is needed, e.g. using data assimilation for parameter estimation.

So-called dual state-parameter estimation uses two separate filters to estimate state variables on the one side and model parameters on the other side. A suitable coupling method linking both filters has to be added [125].

A much more straightforward approach is joint state-parameter estimation. The state vector is augmented, i.e. the vector of parameters is concatenated to the state vector and the assimilation procedure as described in the preceding section is performed [150, 9, 2, 184, 161, 152]. In this way, parameter and state estimation are coupled in a more natural way and the parameter uncertainty is directly included into the state forecast. A difficulty in this context is the specification of the correlation between the state error covariance matrix and the parameter error covariance matrix.

In the following, the general proceeding of state augmentation in context of joint state-parameter estimation is presented, before it is applied for source estimation.

State Augmentation

The aim is to estimate n_θ parameters θ_i, $i \in \{1, \ldots, n_\theta\}$, along with the state vector $\mathbf{x}$. Basis of the joint estimation method is the augmentation of the state vector. Thus, the parameter vector $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ is concatenated to the state vector yielding the augmented state vector

$$\tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ \boldsymbol{\theta} \end{pmatrix} \in \mathbb{R}^{n+n_\theta} \tag{2.55}$$

The data assimilation methods presented before should now be used with the augmented state vector. To attain this, a number of further arrangements have to be made.

First, one has to consider the dynamics of the parameters. While the parameters do not have to be assumed to be constant over time, their dynamics are not known. Assuming that the parameters are

only slightly changing in time, their dynamics can be approximated by $\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)}$ and the forecast equation for the augmented state vector yields

$$\tilde{\mathbf{x}}^{(k+1)} = \begin{pmatrix} \mathbf{M}\left(\boldsymbol{\theta}^{(k+1)}\right)\mathbf{x}^{(k)} \\ \boldsymbol{\theta}^{(k)} \end{pmatrix} + \begin{pmatrix} \mathbf{f}^{(k+1)}\left(\boldsymbol{\theta}^{(k+1)}\right) \\ \mathbf{0} \end{pmatrix}. \tag{2.56}$$

The augmented model error made in this step

$$\tilde{\boldsymbol{\eta}} = \begin{pmatrix} \boldsymbol{\eta} \\ \boldsymbol{\eta}_{\boldsymbol{\theta}} \end{pmatrix} \tag{2.57}$$

consists of the state model error and the parameter model error $\boldsymbol{\eta}_{\boldsymbol{\theta}}$ with corresponding model error covariance matrix

$$\tilde{\mathbf{Q}} = \begin{pmatrix} \mathbf{Q} & \mathbf{Q}_{\mathbf{x}\theta} \\ \mathbf{Q}_{\mathbf{x}\theta}^T & \mathbf{Q}_{\theta} \end{pmatrix}. \tag{2.58}$$

While the parameter error covariance matrix $\mathbf{Q}_{\theta}$ states how much uncertainty is added to the parameters during forecast, the cross-covariance matrix $\mathbf{Q}_{\mathbf{x}\theta}$ describes the correlation between the parameter error and the state model error. A similar matrix is set up for the augmented error covariance matrix

$$\tilde{\mathbf{P}} = \begin{pmatrix} \mathbf{P} & \mathbf{P}_{\mathbf{x}\theta} \\ \mathbf{P}_{\mathbf{x}\theta}^T & \mathbf{P}_{\theta} \end{pmatrix}. \tag{2.59}$$

The parameter error covariance $\mathbf{P}_{\theta}$ describes the quality of the parameter estimates, whereas the cross-covariance matrix $\mathbf{P}_{\mathbf{x}\theta}$ represents the correlation between state error and parameter error. Linearizing (2.56) results in the error covariance forecast equation

$$\tilde{\mathbf{P}}^{(k+1)} = \tilde{\mathbf{M}}\tilde{\mathbf{P}}^{(k)}\tilde{\mathbf{M}}^T + \tilde{\mathbf{Q}}^{(k)} \tag{2.60}$$

with

$$\tilde{\mathbf{M}} = \begin{pmatrix} \mathbf{M}\left(\boldsymbol{\theta}^{(k+1)}\right) & \left(\dfrac{\partial \mathbf{M}}{\partial \boldsymbol{\theta}} + \dfrac{\partial \mathbf{f}}{\partial \boldsymbol{\theta}}\right)_{\theta=\theta^{(k+1)}} \\ \mathbf{0} & \mathbb{1} \end{pmatrix}. \tag{2.61}$$

Parameters are usually not measured directly. For this reason, the augmented observation model matrix has the simple form

$$\tilde{\mathbf{H}} = \begin{pmatrix} \mathbf{H} & \mathbf{0} \end{pmatrix}. \tag{2.62}$$

With this, all augmented entities have been defined and the standard data assimilation methods can be used with the augmented state vector to jointly estimate state and parameters.

Joint State-Parameter Estimation for Source Estimation

Many ways have been presented to estimate source functions and locations based on a bunch of measurements [145]. In this work, the source should be estimated using state augmentation and joint state-parameter estimation as previously described. Estimating the background velocity and diffusivity along with the state vector is also possible, but it should be obeyed that nonlinearity would be inserted into the system.

The source $s(\mathbf{r}, t)$, or the right hand side of the advection-diffusion equation (2.2) is assumed to consist of a time-varying output function $\theta(t)$ and a spatial function $q(\mathbf{r})$ describing the location of the source

$$s(\mathbf{r}, t) = \theta(t) q(\mathbf{r}). \tag{2.63}$$

In the presence of multiple sources, this expression can be extended to

$$s(\mathbf{r}, t) = \sum_{j=1}^{n_\theta} \theta_j(t) q_j(\mathbf{r}). \tag{2.64}$$

At first, it is assumed that the source locations $q_j(\mathbf{r})$ are known and that the source output functions $\theta_j(t)$ have to be estimated. This results in the parameter vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{n_\theta})^T$ and, referring to (2.17), (2.33) and (2.56), the linear forecast equation

$$\tilde{\mathbf{x}}^{(k+1)} = \begin{pmatrix} \mathbf{M} & -\mathbf{S} \\ \mathbf{0} & \mathbb{1} \end{pmatrix} \tilde{\mathbf{x}}^{(k)} + \begin{pmatrix} \hat{\mathbf{f}}^{(k+1)} \\ \mathbf{0} \end{pmatrix} \tag{2.65}$$

with

$$\mathbf{S} = \Delta t \int_\Omega s \phi_j \, \mathrm{d}\Omega \quad \text{and} \tag{2.66}$$

$$\hat{\mathbf{f}} = \Delta t D \int_{\partial\Omega_N} t_S \phi_j \, \mathrm{d}\Gamma. \tag{2.67}$$

The different parameter model errors as well as the initial parameter errors can be assumed to be uncorrelated, i.e. $\mathbf{Q}_{\boldsymbol{\theta}}$ as well as $\mathbf{P}_{\boldsymbol{\theta}}^{(0)}$ are diagonal. Assuming that the source just started its emission, the initial cross-covariance matrix $\mathbf{P}_{\mathbf{x}\theta}^{(0)}$ can be approximately set to zero and the model error cross-covariance matrices $\mathbf{Q}_{\mathbf{x}\theta}$ are set to zero as well. Due to propagation of the covariance matrix, e.g. (2.35), the error cross-covariance matrix $\mathbf{P}_{\mathbf{x}\theta}$ deviates from zero and correlation between parameter and state vector is build up. If an ensemble method is used as data assimilation method, the ensembles are perturbed with samples drawn from the parameter model error distribution with diagonal error covariance matrix $\mathbf{Q}_{\boldsymbol{\theta}}$. The perturbation is performed prior to the actual forecast step to represent the influence of the parameter error on the state model.

Using the described augmented state vectors, matrices and parameters together with a data assimilation method results in estimating of the source output parameter θ_j together with the state vector $\mathbf{x}$.

If the complete source term $s(\mathbf{r}, t)$ is unknown, the idea is to approximate the actual source by a linear combination of suitable basis functions $q_j(\mathbf{r})$ with center $\mathbf{r}_j$, i.e.

$$s(\mathbf{r}, t) \approx s_q(\mathbf{r}, t) = \sum_{j=1}^{n_\theta} \theta_j(t) q_j(\mathbf{r}). \tag{2.68}$$

This expression equals (2.64) so that the procedure described above can be also applied when looking for the complete source function. However, this presumes that the source can reasonably be approximated by (2.68) with a relatively low number n_θ of basis functions. This applies especially for scenarios, in which the potential source area is known and the true source function $s(\mathbf{r}, t)$ has large spatial support. Hence, this approach is not feasible for point sources with completely unknown source area. The basis functions $q_j(\mathbf{r})$ to be used in this context can be for example radial basis functions (RBF)

$$q_j(\mathbf{r}) = \exp\left(-a_j \|\mathbf{r} - \mathbf{r_j}\|^2\right) \tag{2.69}$$

with $a_j > 0$ or piecewise polynomial functions with compact support as described in [63].

2.5 Identification of Measurement Locations

The main advantage of the systems considered in this work is that they can adapt their position based on the currente state estimate and its statistics. With this dynamic data-driven behavior, trajectories are generated that provide much more information enabling the chance to obtain a much better estimate. But how can such trajectories be constructed? Which measure divides a good trajectory from a bad trajectory? In order to find optimal control inputs, a sophisticated optimal control problem, which minimizes a measure of the uncertainty subject to vehicle dynamics, process dynamics, uncertainty dynamics, measurement update equations, etc., would have to be solved [142, 187, 187, 188, 162, 182, 82, 199]. However, to account for real-time requirements related to online dynamic data-driven methods, the trajectory generation has to be computationally very efficient. For this reason, the solution of a monolithic optimal control problem is waived and, to gain computational efficiency, the identification of discrete measurement locations, which are called target points in the following, is proposed instead. It is then the task of a vehicle controller (see Section 2.6) to guide the sensor vehicles to the generated target points.

As the main objective is to reduce the uncertainty of the estimates, the target points introduced in Subsection 2.5.1 are partially based on the current error covariance matrix. A new sequential algorithm avoiding clustering of target points is developed that is based on the error covariance matrix and the estimation method used for data assimilation. To also account for the goal of identification of the source location, source-based target locations are identified as well. Furthermore, local appealing points are introduced with the intention to pull the vehicle trajectories towards interesting side locations (see Subsection 2.5.2).

2.5.1 Target Points

Target points are locations at which an additional measurement seems to be suitable in terms of information gain. The suitability of an observation at a specific location can be expressed in multiple ways. Examples are the use of singular vectors [24], sensitivities [39], or mutual information [71, 34].

In this work, two types of target points are considered. First, locations related to a maximum error variance are regarded. These locations are identified on the basis of the current error covariance matrix, which is provided by the data assimilation method. In this way, the uncertainty of the estimates should be reduced. Second, target points corresponding to source locations are used. Placing sensors at the expected position of the source not only is the best practice for estimating the source output, also it helps to reduce the overall system's uncertainty since a precise source output is known.

Although target points are only optimal at the instance of time they are generated and their optimality might decrease over time, target points are not recalculated in every time step in order to save computa-

tional time and to prevent target points from jumping from one position to another. Instead, covariance based target points are moved in wind direction with wind velocity to account for the temporal covariance evolution dominated by the wind.

If one of the following two events takes place, a new target point will be calculated.

1. The target point is reached by a sensor vehicle.
2. The target point has been active for a certain amount of time and has not been reached yet.

Concerning the number of target points, it is a good choice to choose the number of target points larger than the number of sensor vehicles. In this way, every sensor vehicle gets a job to fulfill. However, one should not choose the number of target points much higher than the number of sensor vehicles since, besides the increased amount of computation, some target points will hardly be respected. Hence, the number of target points is set to the number of vehicles in this work.

Covariance-based target points

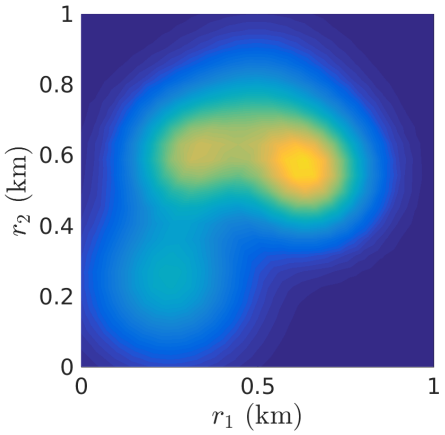
The error covariance matrix provided by the data assimilation method is a suitable measure of the quality of the state estimate. It indicates at which locations the estimate is rather uncertain, at which the estimate is rather reliable and how the estimates at specific locations relate to each other. The larger the entries in the error covariance matrix, the possibly higher the deviations between true state and estimate. Therefore, the objective is to measure at locations that maximally reduce the entries in the error covariance matrix.

For matrix minimization, a number of different strategies exist, e.g. A-optimality minimizing the trace of the matrix or D-optimality minimizing its determinant. However, it is hard and time-consuming to find the location leading to a maximum reduction of the error covariance matrix in one of the above senses. A high-dimensional optimization problem would have to be solved to obtain the solution of this problem. For this reason, an alternative problem is solved that does not consider the effects of the cross-correlations and simply tries to reduce the corresponding diagonal matrix $\mathbf{P_D} = \text{diag}(\mathbf{P})$, only depending on the error variances.

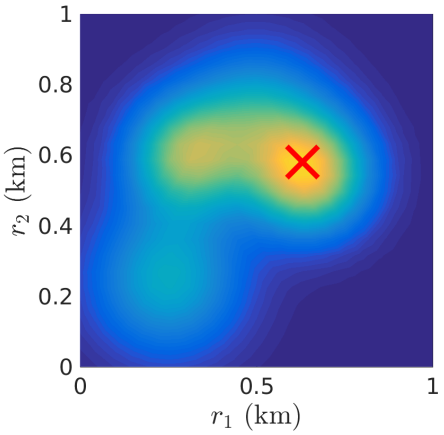
If such a diagonal matrix is considered, one can simply choose the locations belonging to the highest entries of the diagonal of the covariance matrix as target points. Measuring at these positions would reduce the trace and determinant of the diagonal matrix $\mathbf{P_D}$ in the maximum way. However, this solution would not lead to a balanced distribution of target points in most cases. It is very likely that the target points would be clustered at a specific location where the error variance is at its maximum. Locations farther away corresponding to potentially only a slightly less error variance would not be considered.

To circumvent this clustering effect, an iterative algorithm has to be set up. The first target point will be the location corresponding to maximum error variance $\mathbf{P_D}$. Now, the analysis error covariance matrix $\mathbf{P^a}$ is calculated using (2.37) pretending that a measurement at the just selected target location is performed. In this way, the impact of a measurement corresponding to the new target point has been taken into account. The new error variance $\mathbf{P_D^a}$ is the basis for choosing the second target point. After that, the covariance matrix is recomputed again and the procedure is repeated until all target points have been found. The procedure is summarized in Algorithm 1 and illustrated in Figure 2.8.

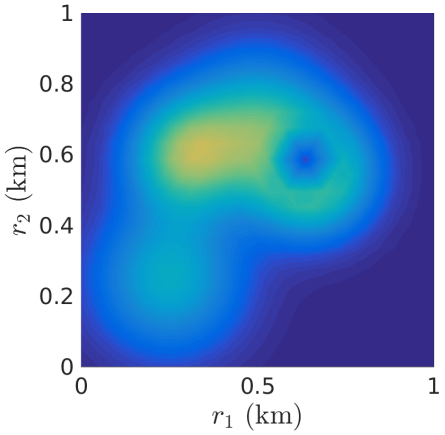
Please note that the algorithm also works when using an EnSRF and that the full error covariance matrix does not have to be constructed in this case. Instead, the vector $\mathbf{v}$ with $(\mathbf{v})_j = \sum_{i=1}^{n_s} (\mathbf{X})_{j,i}^2$ represents the error variance and the deviation matrix $\mathbf{X}$ for selecting the target points is updated using (2.46).



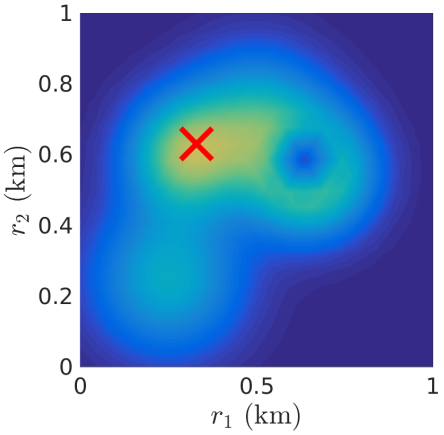
(a) Set up current error variance field considering all existing target points.



(b) Pick location belonging to maximum error variance as target point.



(c) Update error variance pretending measurement at picked target point is available.



(d) Pick location belonging to maximum error variance as target point and proceed.

Figure 2.8: Illustration of the procedure for identifying target points

Source-based target points

If the source is unknown and estimated jointly with the process state, one should also consider target points that take the source estimate into account. As a high amount of uncertainty is introduced to the model at the source location, especially when the source output function is highly dynamical, these locations are a good choice for a target point. A sensor at the source position provides a higher quality estimate of the source output function $\theta(t)$ resulting in less uncertainty introduced to the state model.

If the source location is not exactly known, the location belonging to the maximum value of the approximated source function $s_q(\mathbf{x}, t)$ should be chosen as a target point. This position provides the best estimate for the center of the source. If several source target points are selected, the source target points can be determined in a sequential procedure similar to the covariance-based target points. Instead of performing an analysis after each identification, an RBF with center and amplitude corresponding to the

currently identified target position is subtracted from the approximated source function $s_q(\mathbf{x}, t)$ to avoid clustering.

Algorithm 1 shows the complete procedure for generating the target points in each step.

Data: Current source target point set M_s, current uncertainty-based target point set M_u, current source contribution $\mathbf{s_q}$, RBF $\mathbf{f}(\mathbf{r})$, current observation matrix $\mathbf{H}$, current deviation matrix $\mathbf{X}$, observation error covariance matrix R
Result: Updated target point sets M_s and M_u
$M_s^o \leftarrow$ find outdated source target points in M_s;
$M_u^o \leftarrow$ find outdated and occupied uncertainty-based target points in M_u;
$M_s \leftarrow M_s \setminus M_s^o$;
$M_u \leftarrow M_u \setminus M_u^o$;
if $M_s^o \neq \emptyset$ **then**
  for $\mathbf{r}_i \in M_s$ **do**
    $\mathbf{s_q} \leftarrow \mathbf{s_q} - \mathbf{H}_i \mathbf{s_q} \mathbf{f}(\mathbf{r}_i)$;
  end
  for $i \leftarrow 1$ **to** $|M_s^o|$ **do**
    $s_{max} \leftarrow \max(\mathbf{s_q})$;
    $\mathbf{r}_i \leftarrow$ location belonging to $\max(\mathbf{s_q})$;
    $M_s \leftarrow M_s \cup \{\mathbf{r}_i\}$;
    $s_q \leftarrow \mathbf{s_q} - s_{max} \mathbf{f}(\mathbf{r}_i)$;
  end
end
if $M_u^o \neq \emptyset$ **then**
  $\tilde{\mathbf{H}} \leftarrow$ set up observation matrix corresponding to target points M_s and M_u;
  $\tilde{\mathbf{X}} \leftarrow$ EnSRF($\mathbf{X}, \mathbf{R}, \tilde{\mathbf{H}}$) (deviation matrix $\mathbf{X}$ if analysis is performed with $\mathbf{R}$ and $\tilde{\mathbf{H}}$);
  for $i \leftarrow 1$ **to** $|M_u^o|$ **do**
    $\mathbf{v} \leftarrow$ set up variance with $(\mathbf{v})_j = \sum_{i=1}^{n_s} (\mathbf{X})_{j,i}^2$, $j = \{1, \ldots, n\}$;
    $\mathbf{r}_i \leftarrow$ location belonging to $\max(\mathbf{v})$;
    $M_u \leftarrow M_u \cup \{\mathbf{r}_i\}$;
    $\tilde{\mathbf{H}} \leftarrow$ set up observation matrix corresponding to new target point $\mathbf{r}_i$;
    $\tilde{\mathbf{X}} \leftarrow$ EnSRF($\mathbf{X}, \mathbf{R}, \tilde{\mathbf{H}}$) (deviation matrix $\mathbf{X}$ if analysis is performed with $\mathbf{R}$ and $\tilde{\mathbf{H}}$);
  end
end

Algorithm 1: Sequential procedure to identify new informative target points

2.5.2 Local Appealing Points

With the target points, a set of locations has been identified that indicates useful additional measurements positions. However, the sensors measure at every time step so that a lot of information can also be gathered on the way to the target points. The choice of the trajectory leading from the current position to the target point is, thus, also decisive.

To account for a trajectory to the target point that maximizes information and reduces uncertainty, local appealing points are introduced. Local appealing points are located near the current position of the sensor

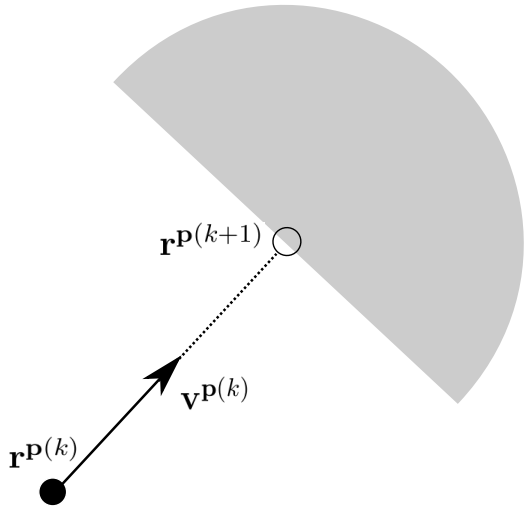


Figure 2.9: Feasible region for a local appealing point. The grey semicircle depicts the area, in which a suitable local appealing point is determined.

vehicle and try to attract the sensor vehicle to locations that possess a higher error variance providing more informative measurements.

In every step, such a local appealing point is calculated for every sensor vehicle. To preserve the forward motion of the sensor vehicle, the appealing point lies within the semicircular environment in direction of the sensor vehicle's velocity as depicted in Figure 2.9. For this environment, the location belonging to the maximum error variance $\mathrm{diag}(\mathbf{P})$ as described in the previous subsection is determined as local appealing point.

Minimizing both, the distances to the global targets and to the local, individual appealing points results in a trajectory of the sensor vehicle from its starting location to the target point, minimizing uncertainty on its way.

2.6 Cooperative Vehicle Controller

The purpose of the vehicle controller to be applied is to guide the sensor vehicles to a set of target points, which have been previously identified to be a suitable measurement location. Although the number of target points corresponds to the number of sensor vehicles, no target location is assigned to a specific sensor vehicle. Instead, it is the common task of the sensor vehicles to cooperatively reach as many target points as possible or at least to minimize the distance to them.

The cooperation aspect has the benefit that the workload can be dynamically distributed and adapted between the sensor vehicles. Approaches to tackle the cooperative multi-vehicle task allocation problems have been presented in literature (see [99] for a review). Besides heuristic approaches, such as behavior-based or market-based methods, optimization-based approaches have been investigated. The latter have the advantage that optimal solutions for the considered cost function subject to the sensor vehicles' dynamics are considered. With the use of a model predictive control method, the resulting optimal control problem receives a feedback control so that a dynamic change in the target point can be taken into account and potential deviations can be compensated. Exemplary optimization and model predictive control-based approaches have been applied in [111, 146].

In this work, the optimization-based cooperative vehicle controller with model-predictive control proposed in [103, 54] is applied with its extensions to the present problem presented in [147, 56]. Based on a linear discrete-continuous optimization scheme, optimal control inputs are calculated in a model predictive control fashion This results in collision-free trajectories and optimal target allocation, all with respect to the dynamics and physical properties of the sensor vehicles.

The basis of the approach is the formulation of the multi-vehicle system in a Mixed Logical Dynamical framework as presented in Subsection 2.6.1. The formulation is used to set up an optimal control problem (see Subsection 2.6.2) that is solved via Model Predictive Control (see Subsection 2.6.3).

2.6.1 MLD Model for the Multi-Vehicle System

The basis of the applied cooperative vehicle controller is a Mixed Logical Dynamical (MLD) formulation of the considered multi-vehicle system. The MLD framework originally proposed in [14] has been designed to model and control constrained linear systems containing logical rules. In the context of this work, motion dynamics, distances and target allocation all have to be available in an MLD representation. For this reason, a number of linearizations are necessary [103, 54].

Motion Dynamics

In an MLD framework, the dynamics of the multi-vehicle system must be available in a linearized form

$$\mathbf{p}_i^{(k+1)} = \mathbf{A}_i^{\mathbf{p}} \mathbf{p}_i^{(k)} + \mathbf{B}_i^{\mathbf{p}} \mathbf{u}_i^{(k)} \,, \tag{2.70}$$

where $\mathbf{p}_i^{(k)}$ and $\mathbf{u}_i^{(k)}$ are state and control input of vehicle $i \in \{1, \ldots, n_V\}$ at time $t^{(k)}$ with global vehicle state $\mathbf{p}_\mathbf{g}^{(k)} = \text{col}\{\mathbf{p}_i^{(k)}\}_{i=1}^{n_V}$ and $\mathbf{u}_\mathbf{g}^{(k)} = \text{col}\{\mathbf{u}_i^{(k)}\}_{i=1}^{n_V}$. The considered vehicle dynamics (2.29) are already available in a linear form so that no further linearization is necessary.

Besides the vehicles, it has been described in the previous section that the target points partially also undergo a movement caused by the wind velocity so that the linearized dynamics of the target point state $\mathbf{q}_i^{(k)}$, $i \in \{1, \ldots, n_M\}$ is described by

$$\mathbf{q}_i^{(k+1)} = \mathbf{A}_i^{\mathbf{q}} \mathbf{q}_i^{(k)} + \mathbf{b}_i^{\mathbf{q}}. \tag{2.71}$$

The global target point state vector yields $\mathbf{q}_\mathbf{g}^{(k)} = \text{col}\{\mathbf{q}_i^{(k)}\}_{i=1}^{n_M}$.

Distances

In the optimization problem to be formulated, distances between vehicles and target points, between vehicles and local appealing points and between vehicles and vehicles occur. As the Euclidean distance is unsustainable in a linear problem, an alternative formulation has to be set up. A linear approximation of the distance $d_{ij}^{(k)}$ between a vehicle i and a target point j is, for example, obtained by the inequalities

$$(r_{1,i}^{p(k)} - r_{1,j}^{q(k)}) \sin \frac{2\pi\gamma}{n_\gamma} + (r_{2,i}^{p(k)} - r_{2,j}^{q(k)}) \cos \frac{2\pi\gamma}{n_\gamma} \;\leq\; d_{ij}^{(k)} \,, \tag{2.72}$$

where $(r_{1,i}^{p(k)}, r_{2,i}^{p(k)})^T$ denotes the position of vehicle i, $(r_{1,i}^{q(k)}, r_{2,i}^{q(k)})^T$ the position of target j, and $\gamma = \{1, \ldots, n_\gamma\}$. The minimum value $d_{ij}^{(k)}$ fulfilling all inequalities (2.72) is an approximation of

the Euclidean distance $d_{ij}^{(k)} \approx \sqrt{(r_{1,i}^{p(k)} - r_{1,i}^{q(k)})^2 + (r_{2,i}^{p(k)} - r_{2,i}^{q(k)})^2}$. An increase of the constant parameter $n_\gamma \in \mathbb{N}$ improves the accuracy coevally increasing the computational complexity.

Measurement Constraints

To indicate if a target point j is occupied by a vehicle i, the binary variable $b_{ij}^{(k)} \in \{0, 1\}$ is introduced for each pair (i, j). It equals one if the distance between target point j and vehicle i is smaller than a predefined threshold d_{meas} and zero otherwise:

$$b_{ij}^{(k)} = 1 \quad \Leftrightarrow \quad d_{ij}^{(k)} \leq d_{meas} \,, \tag{2.73}$$

This relation can be represented in a linear manner by the inequalities

$$d_{ij}^{(k)} - d_{meas} \leq M_1(1 - b_{ij}^{(k)}) \quad \text{and} \tag{2.74a}$$

$$d_{ij}^{(k)} - d_{meas} \geq \varepsilon + (m_1 - \varepsilon) b_{ij}^{(k)} \,, \tag{2.74b}$$

with $M_1 \geq \max\{d_{ij} - d_{meas}\}$, $m_1 \leq \min\{d_{ij} - d_{meas}\}$, and a small tolerance ε.
Furthermore, a variable $s_j^{(k)} \in \{0, 1\}$ is used to indicate if a target point j is occupied or not, i.e.

$$s_j^{(k)} = 0 \quad \Leftrightarrow \quad \sum_{i=1}^{n_V} b_{ij}^{(k)} \geq 1 \,. \tag{2.75}$$

This relation can be linearized by the inequalities

$$1 - \sum_{i=1}^{n_V} b_{ij}^{(k)} \leq M_2 \cdot s_j^{(k)} \quad \text{and} \tag{2.76a}$$

$$1 - \sum_{i=1}^{n_V} b_{ij}^{(k)} \geq \varepsilon + (m_2 - \varepsilon)(1 - s_j^{(k)}) \tag{2.76b}$$

with $M_2 \geq \max\{1 - \sum b_{ij}\} = 1$ and $m_2 \leq \min\{1 - \sum b_{ij}\}$.
Another important entity is the product $h_{ij} \in \mathbb{R}$ of the target dependent variable s_j and the target and vehicle dependent variable d_{ij}

$$h_{ij}^{(k)} = s_j^{(k)} \cdot d_{ij}^{(k)} \,. \tag{2.77}$$

It represents the distance of vehicle i to target point j if the target point is not yet observed and zero otherwise. In linearized form, (2.77) reads

$$h_{ij}^{(k)} \leq M_3 \cdot s_j^{(k)} \,, \tag{2.78a}$$

$$h_{ij}^{(k)} \leq d_{ij}^{(k)} \,, \quad \text{and} \tag{2.78b}$$

$$-h_{ij}^{(k)} \leq -d_{ij}^{(k)} + M_3(1 - s_j^{(k)}) \,, \tag{2.78c}$$

with $M_3 \geq \max\{d_{ij}\}$.

Collision Avoidance

To avoid collisions, every two vehicles i and j, $i \neq j$ should keep a minimum distance d_{min} to each other, i.e. the region

$$(r_{1,i}^{p(k)} - r_{1,j}^{p(k)}) \sin \frac{2\pi\gamma}{n_\gamma} + (r_{2,i}^{p(k)} - r_{2,j}^{p(k)}) \cos \frac{2\pi\gamma}{n_\gamma} \leq d_{min} , \tag{2.79}$$

should not be entered.
This means that at least one of the inequalities (2.79) must be violated. Introducing an additional set of binary variables $b_{ij\gamma}^{(k)}$, this relation can be formulated as

$$(r_{1,i}^{p(k)} - r_{1,j}^{p(k)}) \sin_\gamma + (r_{2,i}^{p(k)} - r_{2,j}^{p(k)}) \cos_\gamma > d_{min} \Rightarrow b_{ij\gamma}^{(k)} = 0 \tag{2.80}$$

$$\text{and} \quad \sum_{\gamma=1}^{n_\gamma} b_{ij\gamma}^{(k)} \leq n_\gamma - 1 . \tag{2.81}$$

resulting in the linearized form

$$(r_{1,j}^{p(k)} - r_{1,i}^{p(k)}) \sin_\gamma + (r_{2,j}^{p(k)} - r_{2,i}^{p(k)}) \cos_\gamma \leq -d_{min} + M_4 \cdot b_{ij\gamma}^{(k)} , \tag{2.82}$$

where $M_4 \geq \max\{(r_{1,j}^{p(k)} - r_{1,i}^{p(k)}) \sin_\gamma + (r_{2,j}^{p(k)} - r_{2,i}^{p(k)}) \cos_\gamma + d_{min}\}$.

2.6.2 Formulation of the Optimal Control Problem

If the multi-vehicle system has been modeled in an MLD framework, the constrained finite time optimal control problem to be solved to get the set of desired control inputs U_{n_T} until final time $T = t^{(n_T)}$ has the general form

$$\min_{U_{n_T}} \ |\mathbf{F}\tilde{\mathbf{p}}^{(n_T)}| + \sum_{k=0}^{n_T-1} |\mathbf{G_1}\tilde{\mathbf{u}}^{(k)}| + |\mathbf{G_2}\boldsymbol{\delta}^{(k)}| + |\mathbf{G_3}\mathbf{z}^{(k)}| + |\mathbf{G_4}\tilde{\mathbf{p}}^{(k)}| \tag{2.83a}$$

$$\text{s.t.} \quad \tilde{\mathbf{p}}^{(k+1)} = \mathbf{A}\tilde{\mathbf{p}}^{(k)} + \mathbf{B_1}\tilde{\mathbf{u}}^{(k)} + \mathbf{B_2}\boldsymbol{\delta}^{(k)} + \mathbf{B_3}\mathbf{z}^{(k)} \tag{2.83b}$$

$$\mathbf{E_2}\boldsymbol{\delta}^{(k)} + \mathbf{E_3}\mathbf{z}^{(k)} \leq \mathbf{E_1}\tilde{\mathbf{u}}^{(k)} + \mathbf{E_4}\tilde{\mathbf{p}}^{(k)} + \mathbf{E_5} . \tag{2.83c}$$

In this general formulation, $\tilde{\mathbf{p}} = (\tilde{\mathbf{p}}_\mathbf{c}, \tilde{\mathbf{p}}_\mathbf{b})^T, \tilde{\mathbf{p}}_\mathbf{c} \in \mathbb{R}^{n_{c1}}, \tilde{\mathbf{p}}_\mathbf{b} \in \{0,1\}^{n_{b1}}$, represents the system state, $\tilde{\mathbf{u}} = (\tilde{\mathbf{u}}_\mathbf{c}, \tilde{\mathbf{u}}_\mathbf{b})^T, \tilde{\mathbf{u}}_\mathbf{c} \in \mathbb{R}^{n_{c2}}, \tilde{\mathbf{u}}_\mathbf{b} \in \{0,1\}^{n_{b2}}$, is the control input, and $\boldsymbol{\delta} \in \{0,1\}^{n_{b3}}$ as well as $\mathbf{z} \in \mathbb{R}^{n_{c3}}$ represent auxiliary binary and continuous vectors, respectively. The solution of the problem is the sequence of optimal control inputs $U_{n_T} = \{\tilde{\mathbf{u}}^{(k)}\}_{k=0}^{n_T-1}$.
In this work, the system state is represented by $\tilde{\mathbf{p}} = (\mathbf{p_g}, \mathbf{q_g})^T$ and the control input is $\tilde{\mathbf{u}} = \mathbf{u_g}$. The objective function (2.83a) is composed of a number of different terms. While the main goal is to lead each vehicle to one of the measurement locations, i.e. to minimize the sum of the binary variables $s_k^{(k)}$, also the distances $h_{ij}^{(k)}$ to unoccupied target points and the distances $d_v^{(k)}$ to the local appealing

points should be minimized - all with minimum control effort $|\mathbf{u}_\mathbf{g}^{(k)}|$. All things considered, the objective function reads

$$\min_{U_{n_T}} \sum_{k=0}^{n_T-1} \left(g_z \sum_{i=1}^{n_V} \sum_{j=1}^{n_M} h_{ij}^{(k)} \; + \; g_\delta \sum_{j=1}^{n_M} s_j^{(k)} \; + \; g_u |\mathbf{u}_\mathbf{g}^{(k)}| \right), \tag{2.84}$$

where g_z, g_δ, $g_u \in \mathbb{R}$ weight the different terms according to their priorities.

The constraint (2.83b) comprises the dynamics described with (2.70) and (2.71). Finally, (2.83c) describes the distance constraint (2.72), the measurement constraints (2.74), (2.76) and (2.78), and the collision avoidance (2.81) and (2.82).

In general, the problem can be extended in a simple way by adding further constraints, e.g. to keep the sensor vehicles in a considered complex problem domain or to avoid obstacles. However, one has to obey that the solution of the problem becomes more time consuming the more constraints are posed to the system and the more sensor vehicles are involved. The problem as described above comprises $n_V n_M(5 + n_\gamma) + 2n_M + \binom{n_V}{2}(n_\gamma + 1)$ linear inequality constraints. With $n_V = n_M$ as chosen in this work, the number of inequality constraints depends quadratically on the number of sensor vehicles used.

2.6.3 MPC Solution of the Optimal Control Problem

Problem (2.83) can be solved in a receding horizon fashion. This means that the problem is not solved over the complete time horizon with final time $T = t^{(n_T)}$ but, starting from the current time $t^{(k)}$, only over a reduced time horizon $n_{Tr}\Delta t$. As solution, one obtains the control sequence $U_{n_{Tr}}^{(k)} := \{\mathbf{u}_\mathbf{g}^{(k+i)}\}_{i=0}^{n_{Tr}-1}$ of control inputs. According to a model-predictive control (MPC) scheme, only the first element of $U_{n_{Tr}}^{(k)}$ is applied to the real vehicle system. Then, a new control sequence is computed based on the control problem considered over the shifted time horizon starting at the new time step $t^{(k+1)}$. In this way, the solution can not only be obtained in a faster way, but this methodology also includes a feedback mechanism since the ability to compensate modeling inaccuracies and perturbations as well as reacting to dynamically changing target points becomes available.

In each step of the MPC procedure, the constrained finite time optimal control problem can be transformed into a Mixed Integer Linear Program. For this kind of problems, robust and efficient computation tools exist that guarantee global optimality without strong dependence on initial guesses.

2.7 Implementation, Evaluation and Results

In this section, the proposed monitoring approach is applied in several monitoring scenarios to show its ability to dynamically estimate the current state and parameter vector and to evaluate important aspects as dependencies on method parameters and computation time. For all scenarios, a twin experiment is conducted. This means that the true solution is represented by the solution from another simulation run. In this way, the estimation error can be obtained at every time and observations can be drawn from the true solution.

Starting with some notions on the implementation in Subsection 2.7.1, Subsection 2.7.2 treats a basic problem, for which the functioning of the monitoring approach is shown. It is demonstrated that the proposed approach performs significantly better than other simple mobile or static strategies. Furthermore, the characteristic behavior of the sensor vehicles is illustrated. In the following, the influence of various

parameters of the monitoring approach on the solution is investigated (see Subsection 2.7.3), before results of joint state-parameter estimation are provided in Subsection 2.7.4. The required numerical effort is investigated in Subsection 2.7.5.

2.7.1 Implementation

The proposed monitoring approach has been implemented in MATLAB. For the process simulation part, a fluid solver of the Nonlinear Optimization Group, TU Darmstadt, has been extended so that the solution of the considered advection-diffusion equation has become possible. The required computation grid is generated with the help of the tool DistMesh [136]. For solving the optimization problem of the vehicle controller, the problem is handed to CPLEX [86] through the MATLAB interface CPLEXINT [179]. The set up of the vehicle control optimization problem was mainly developed by Juliane Euler.

The following computations have all been performed on a computer with Intel (R) Core (TM) i7-3820 3.60 GHz and 16 GB RAM using MATLAB R2015b.

2.7.2 General Results of the Proposed Monitoring Approach

The purpose of the first test case is to show the functionality of the monitoring approach regarding state estimation and to illustrate the behavior of the sensor vehicles. Therefore, a basic test scenario is set up. Please note that the concentration x is treated now and in the following without indicating a corresponding unit. Different choices of the unit would be possible but do not impact the results of this thesis so that the unit of the concentration is ommitted for reasons of simplicity.

The considered problem domain $\Omega = [0\,\mathrm{m}, 1000\,\mathrm{m}] \times [0\,\mathrm{m}, 1000\,\mathrm{m}]$ is depicted in Figure 2.10. The background velocity field is stationary with $w_1(r_1, r_2) = 3 \cdot 10^{-6}\,\mathrm{m^{-1}s^{-1}} \cdot r_2^2$ and $w_2(r_1, r_2) = 2 \cdot 10^{-6}\,\mathrm{m^{-1}s^{-1}} \cdot r_1^2 + 0.5\,\mathrm{ms^{-1}}$ and the constant diffusion coefficient amounts to $D = 5\,\mathrm{m^2s^{-1}}$. A source term is not considered in this scenario – parameter and source estimation are postponed to Subsection 2.7.4.

Instead, the initial condition

$$x^{(0)}(\mathbf{r}) = 0.8 \exp\left(-(\mathbf{r} - \mathbf{r_1})^T \boldsymbol{\Sigma_1} (\mathbf{r} - \mathbf{r_1})\right) + 0.8 \exp\left(-(\mathbf{r} - \mathbf{r_2})^T \boldsymbol{\Sigma_2} (\mathbf{r} - \mathbf{r_2})\right), \tag{2.85}$$

which can be regarded as instantaneous puff, is applied with

$$\boldsymbol{\Sigma_1} = \begin{pmatrix} 4 \cdot 10^{-5} & 4 \cdot 10^{-5} \\ 4 \cdot 10^{-5} & 13 \cdot 10^{-5} \end{pmatrix} [\mathrm{m^{-2}}], \quad \boldsymbol{\Sigma_2} = \begin{pmatrix} 8 \cdot 10^{-5} & 1.5 \cdot 10^{-5} \\ 1.5 \cdot 10^{-5} & 2 \cdot 10^{-5} \end{pmatrix} [\mathrm{m^{-2}}] \tag{2.86}$$

and $\mathbf{r_1} = (350\,\mathrm{m}, 350\,\mathrm{m})^T$, $\mathbf{r_2} = (200\,\mathrm{m}, 220\,\mathrm{m})^T$. The condition serves as the initial condition for the simulation of the true solution $\mathbf{x^t}$ but is not known in the estimation procedure.

Furthermore, the background velocity and the diffusion coefficient of the true solution are perturbed in every step in order to represent a physical model error $\boldsymbol{\eta}$. For the diffusion coefficient, each perturbation step consists of drawing a random variable from a Gaussian distribution with zero mean and $0.1\,\mathrm{m^2s^{-1}}$ standard deviation and adding it to the diffusivity. Both components of the background velocity are uniformly perturbed by random variables also drawn from a Gaussian distribution with zero mean and $0.1\,\mathrm{ms^{-1}}$ standard deviation. A moving average filter over a time horizon of 50 time steps is applied to smooth the velocity perturbations accounting for a more physical behavior. The performed perturbations,

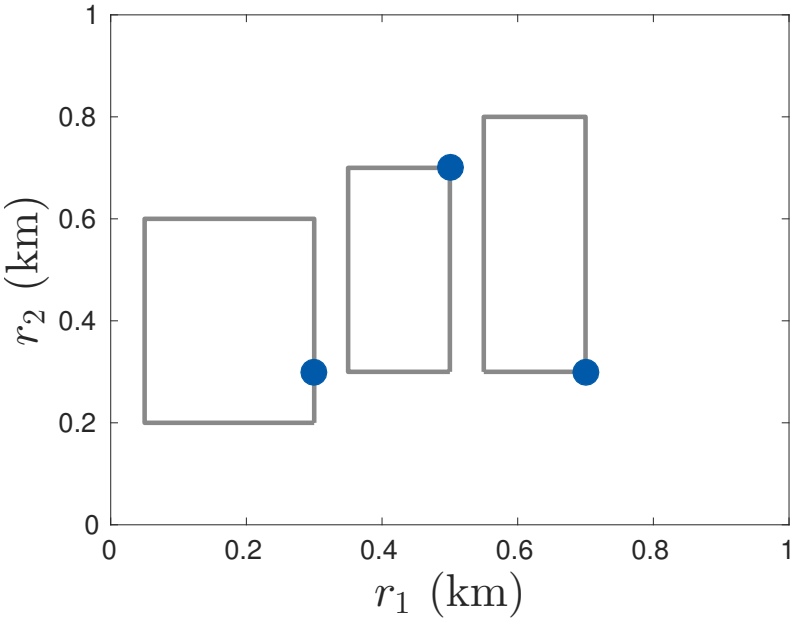


Figure 2.10: Problem domain of the test case scenario. Blue dots mark start positions of sensor vehicles, gray lines indicate lines along which vehicles move in the patrol approach.

while representing actual physical disturbances, do not lead to a model error $\boldsymbol{\eta}$ that is Gaussian as assumed in the derivation of the methods of this work. However, the following results show that even if non-Gaussian, physical model errors are available, good results are obtained.

The initial estimate uses the unperturbed discretization of the advection-diffusion equation (2.33) as forecast equation starting from an initial estimate $x^{(0)}(\mathbf{r}) \equiv 0$. The underlying computation grid consists for this first test case of a relatively low number of $n = 466$ nodes.

Three sensor vehicles, all modeled with double-integrator dynamics (2.29), are applied to gather the measurements. Starting from the locations $\mathbf{r}_1^{\mathbf{p}} = (300\,\mathrm{m}, 300\,\mathrm{m})^T$, $\mathbf{r}_2^{\mathbf{p}} = (500\,\mathrm{m}, 700\,\mathrm{m})^T$ and $\mathbf{r}_3^{\mathbf{p}} = (700\,\mathrm{m}, 300\,\mathrm{m})^T$, their maximum speed amounts to $v_{max}^p = 10\,\mathrm{ms}^{-1}$ and their maximum acceleration is $a_{max}^p = 10\,\mathrm{ms}^{-2}$. As the maximum speed of the background velocity field amounts to approximately $3.9\,\mathrm{ms}^{-1}$, the condition for the sensor vehicle to be faster than the process speed is fulfilled.

The EnSRF is used as data assimilation method. The observation error variance of all sensor vehicles is dependent of the actual amount that is measured and has the form $R_i = R_A \max(y_i, R_{\min})$. For the initial error covariance matrix $\mathbf{P}^{(0)}$ and the basic model error covariance matrix $\boldsymbol{\Psi}$ from (2.52) the following choices are made

$$(\mathbf{P}^{(0)})_{i,j} = P^{(0)}(\mathbf{r}_i, \mathbf{r}_j) = P_A \exp\left(-\frac{\|\mathbf{r}_i - \mathbf{r}_{\mathbf{P}}\|^2}{L_{P1}}\right) \exp\left(-\frac{\|\mathbf{r}_j - \mathbf{r}_{\mathbf{P}}\|^2}{L_{P1}}\right) \exp\left(-\frac{\|\mathbf{r}_i - \mathbf{r}_j\|^2}{L_{P2}}\right) \tag{2.87}$$

$$(\boldsymbol{\Psi})_{i,j} = \Psi(\mathbf{r}_i, \mathbf{r}_j) = Q_A \exp\left(-\frac{\|\mathbf{r}_i - \mathbf{r}_j\|^2}{L_Q}\right). \tag{2.88}$$

Using an ensemble variant, both covariances are transformed into an ensemble set using the eigenvalue decomposition described in Section 2.4.3. In every forecast step, the state-dependent error covariance

Table 2.1: Simulation and controller settings

Ensemble size n_s	20
Observation variance magnitude R_A	0.02
Minimum observation variance R_{min}	10^{-3}
Forecast covariance magnitude P_A	1
Forecast covariance center $\mathbf{r_P}$ [m]	$(250, 250)^T$
Forecast covariance area L_{P1} [m^2]	$1.3 \cdot 10^5$
Forecast covariance area L_{P2} [m^2]	$6 \cdot 10^4$
Model covariance magnitude Q_A	0.05
Model covariance area L_Q [m^2]	$6 \cdot 10^4$
Localization function parameter c_G [m]	200
MPC horizon n_{Tr}	20
Number of hyperplanes n_γ	8

matrix is added to the ensemble according to (2.54). Localization is also used by applying (4.10) of [63] as localization function with parameter c_G. The data assimilation parameters and further control settings are summarized in Table 2.1.

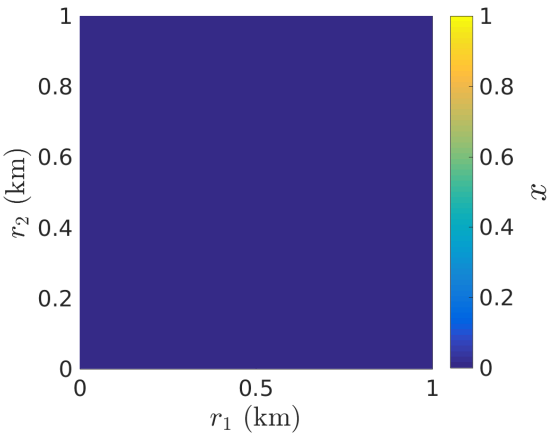
The simulation is conducted over $n_T = 200$ time steps with a time step size of $\Delta t = 2\,\text{s}$. As a result, the mean concentration field is obtained. It is depicted at several times in Figure 2.11 and compared with the true solution at the same times. Starting with a rather bad estimate, the estimate is improved over time. At final time, qualitative differences between the true solution and the estimate can be hardly noticed so that the estimate appears to be rather good.

To allow for a better quantification of the performance of the monitoring strategy, the root mean square error between the true state vector and the estimated state vector has to be determined. As the true solution is known, the root mean square error

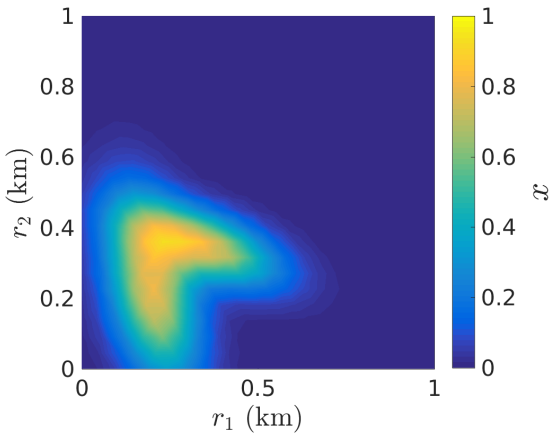
$$\text{rmse}^{(k)} = \frac{\|\mathbf{x}^{\mathbf{t}(k)} - \mathbf{x}^{(k)}\|}{\sqrt{n}} \tag{2.89}$$

can be easily computed and evaluated at every time step $t^{(k)}$. Furthermore, to compensate the potential effect of the involved random variables, the same scenario is run several times and a mean over all simulation runs is taken. The resulting mean error over time is plotted in Figure 2.12. After a strong decrease in error at the beginning, where especially the effects of the initial condition are captured, the error reaches a specific level that is only slightly decreased with advancing time. Due to the limited number of sensor vehicles and the continuous perturbation due to model errors, the true solution, naturally, cannot be reached completely and a specific deviation will always remain. However, despite the continuous perturbation, an increase of the error cannot be noticed so that the quality of the state estimate can be kept relatively constant over time.

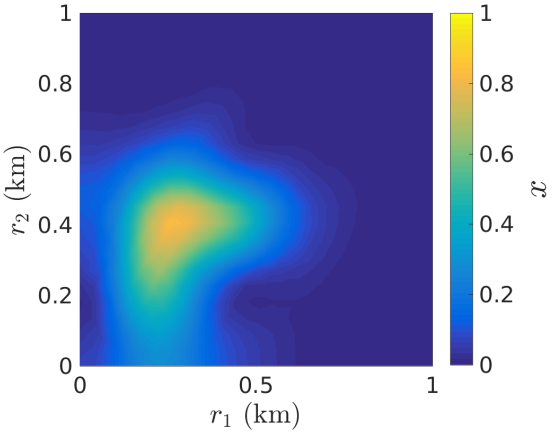
To show that the presented dynamic data-driven approach outperforms other simple strategies, the error evolution is also shown for two other monitoring strategies in Figure 2.12. A static network consisting of three intelligently placed sensors at $\mathbf{r^P}_1 = (200\,\text{m}, 300\,\text{m})^T$, $\mathbf{r^P}_2 = (400\,\text{m}, 500\,\text{m})^T$ and $\mathbf{r^P}_3 = (600\,\text{m}, 600\,\text{m})^T$ can, of course, not match up with the performance of the dynamic data-driven method. While the error reduction at the very beginning is higher (this is due to the fact that the initial position of



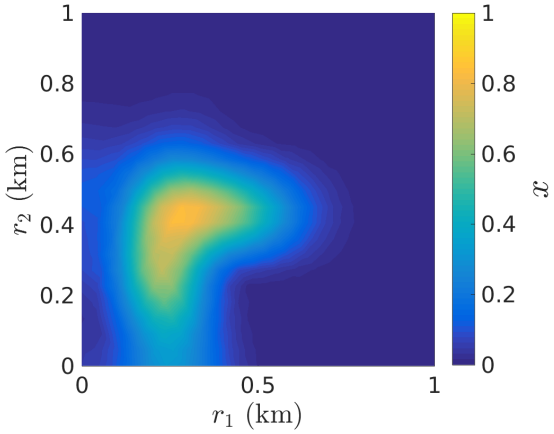
(a) Estimated concentration at $t = 0s$



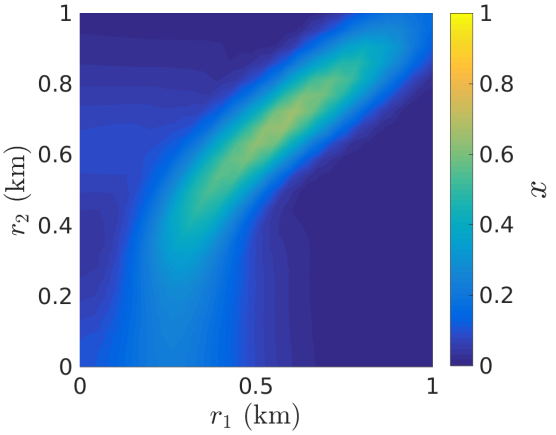
(b) True concentration at $t = 0s$



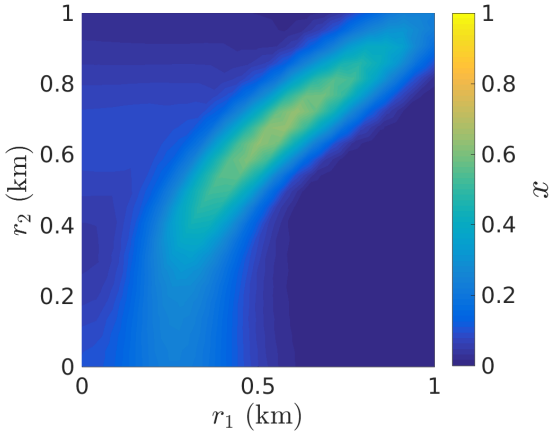
(c) Estimated concentration at $t = 100s$



(d) True concentration at $t = 100s$



(e) Estimated concentration at $t = 400s$



(f) True concentration at $t = 400s$

Figure 2.11: Concentration distribution at different times – estimate (left) vs. true state (right)

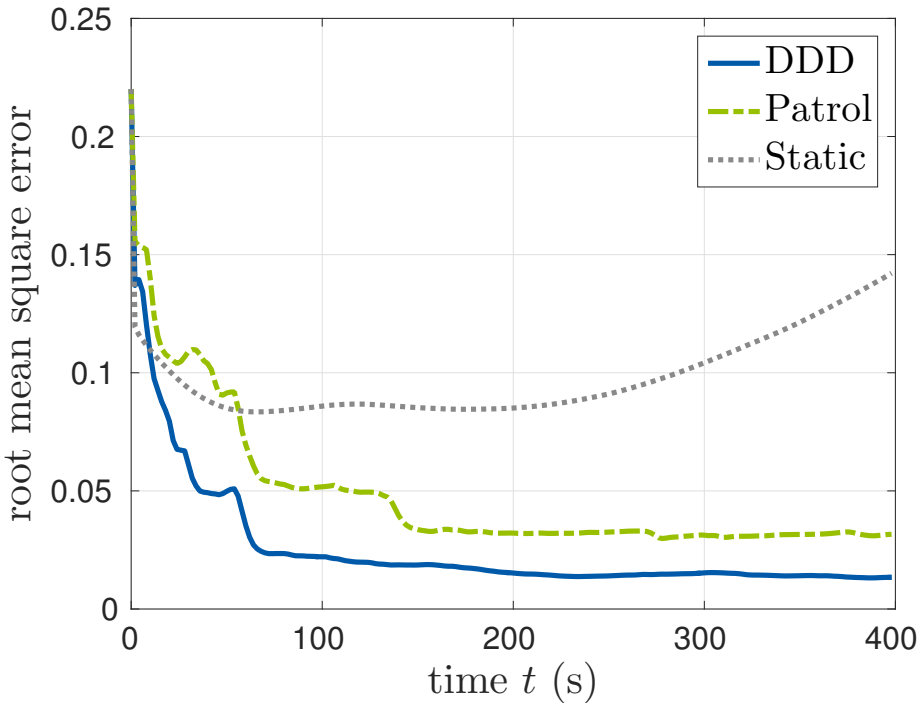


Figure 2.12: Root mean square error of the state estimate obtained with the proposed dynamic data-driven approach, a patrol strategy and a static sensor network over time

the static network is more informative), the data-driven approach soon underbids the error obtained with the static sensor network. With advancing time, the deviation between static estimate and true state even increases so that the performance can be described to be much worse and the motivation to use mobile sensor vehicles in this scenarios can be confirmed.

Another strategy comprises the use of mobile sensor vehicles that patrol along certain, prespecified paths. For the considered patrol approach, the paths are depicted in Figure 2.10. The paths run through regions in which a high amount of information is expected prior to the application. Regarding the error plot in Figure 2.12, the patrol approach also succeeds in keeping the error relatively constant after a first large reduction of the initial error. However, the results of the patrol approach are still far away from the results obtained with the dynamic data-driven approach – the error is nearly twice as high.

An entity that is closely related to the mean error is the error variance. The evolution of the total error variance, i.e. trace$(\mathbf{P})$, is depicted in Figure 2.13. While the static approach does not succeed in combatting the variance increase, leading to an increase in error as seen above, the patrol approach appears not to allow the variance to exceed a certain threshold. The variance slightly increases over a certain time horizon and is rapidly decreased over a short time horizon as the sensor vehicles are sometimes located at informative measurement locations and sometimes not. The dynamic data-driven approach leads to the largest variance reduction since variance reduction is its defined goal. Consequently, the error in the dynamic data-driven approach is the lowest.

The trajectories of the sensor vehicles in the proposed approach is illustrated in Figure 2.14. Thanks to cooperation among the sensor vehicles, each vehicle moves straight to a different target point so that all target points are reached after some time. Every time a target point is reached (or outdated), a new target point is calculated and the sensor vehicles adapt their movement.

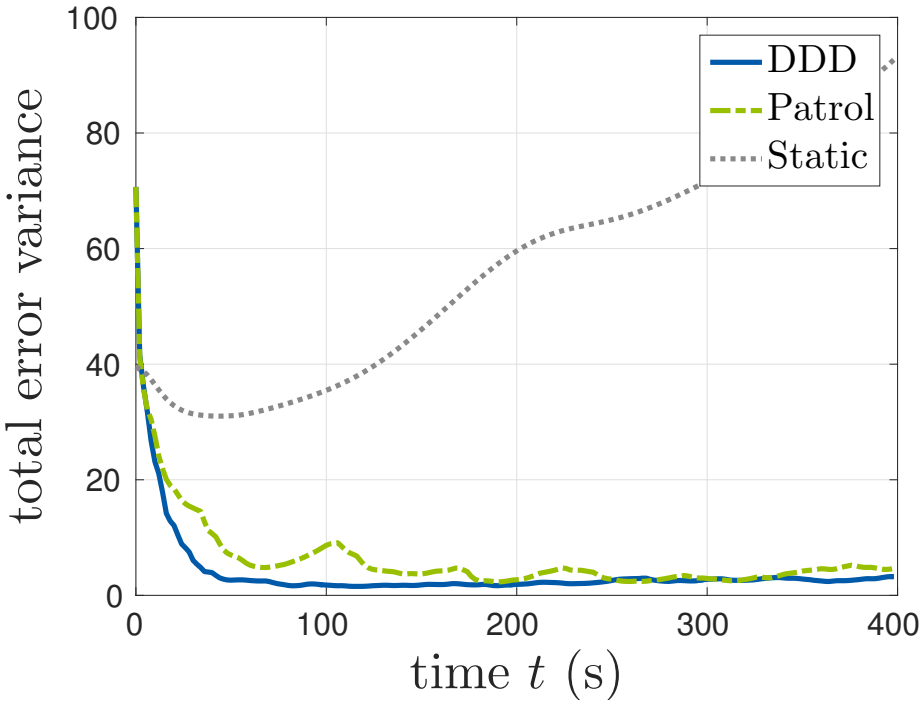
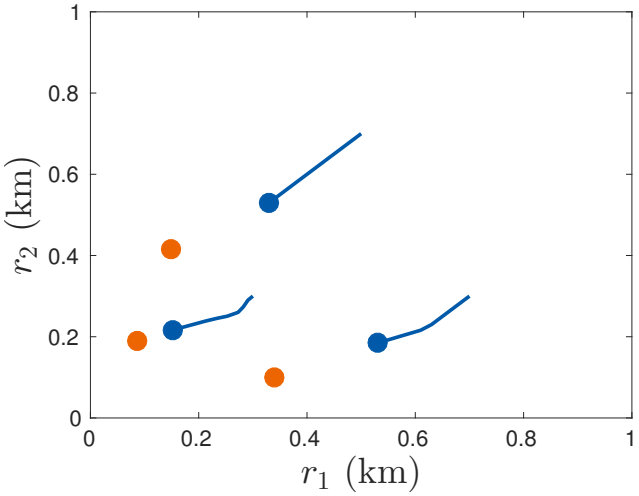
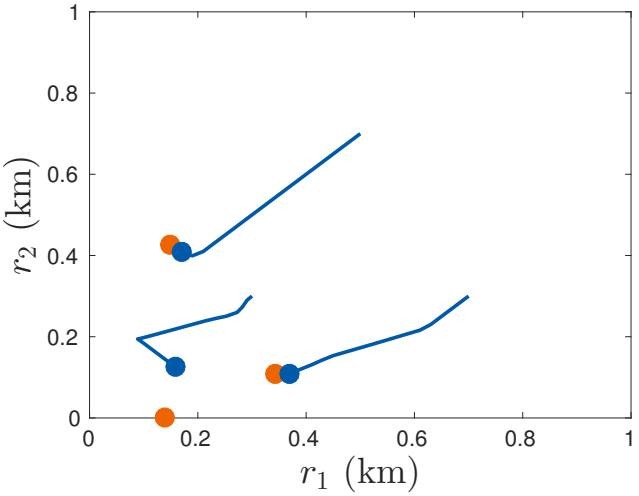


Figure 2.13: Sum of error variance for the proposed dynamic-data driven approach, a patrol strategy and a static sensor network over time



(a) $t = 20s$: Sensor vehicles (blue bullets) are heading for the target points (red bullets). Due to cooperation, each vehicle tries to reach a different target point. The past way is illustrated with a blue line.



(b) $t = 36s$: The vehicle in the left corner has reached a target point and has changed direction to reach the newly calculated target point at the bottom. The other vehicles are still in process of getting to the first generation target points.

Figure 2.14: Illustration of vehicle movement

2.7.3 Investigation of Parameters

The dynamic data-driven approach depends on a number of parameters related to the methods that are employed. In this subsection, the influence of some of these parameters is examined.

Influence of Ensemble Size

At first, the influence of the applied data assimilation method is investigated using the test case setting of the previous subsection. As described in Section 2.4.3, the error covariance matrix of the Kalman Filter can be approximated by the ensemble set of the ensemble filters. But how many ensemble members are necessary to obtain comparable results in the considered application? To answer this question, the monitoring approach is tested with the Kalman Filter as well as with the EnSRF with different ensemble sizes $n_s \in \{5, 20, 50, 100\}$. The error between the true state and the estimate obtained with the monitoring approach with the respective data assimilation method is calculated and plotted over time in Figure 2.15.

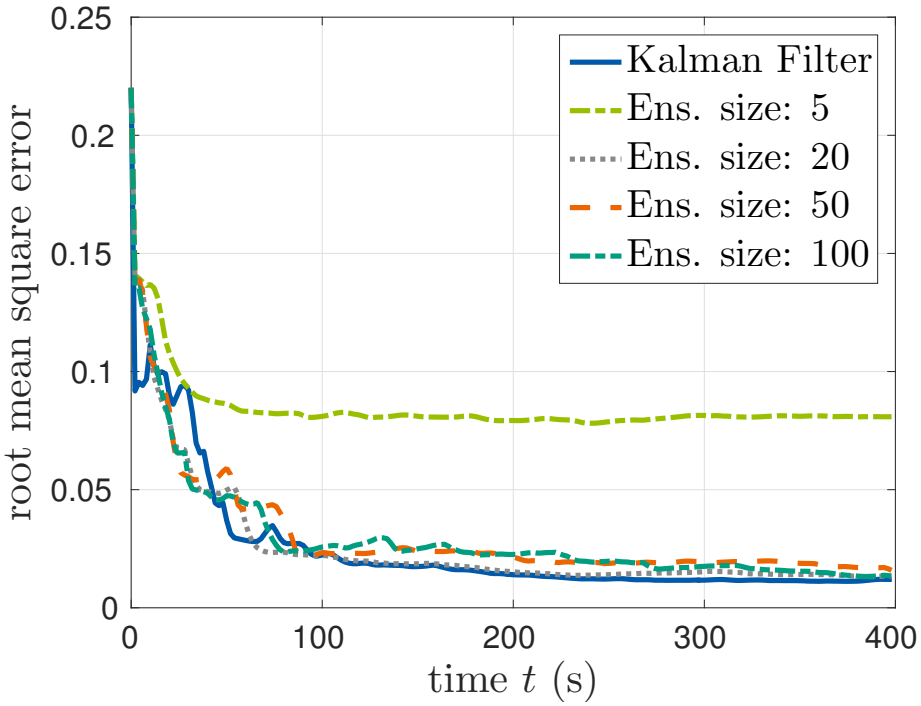


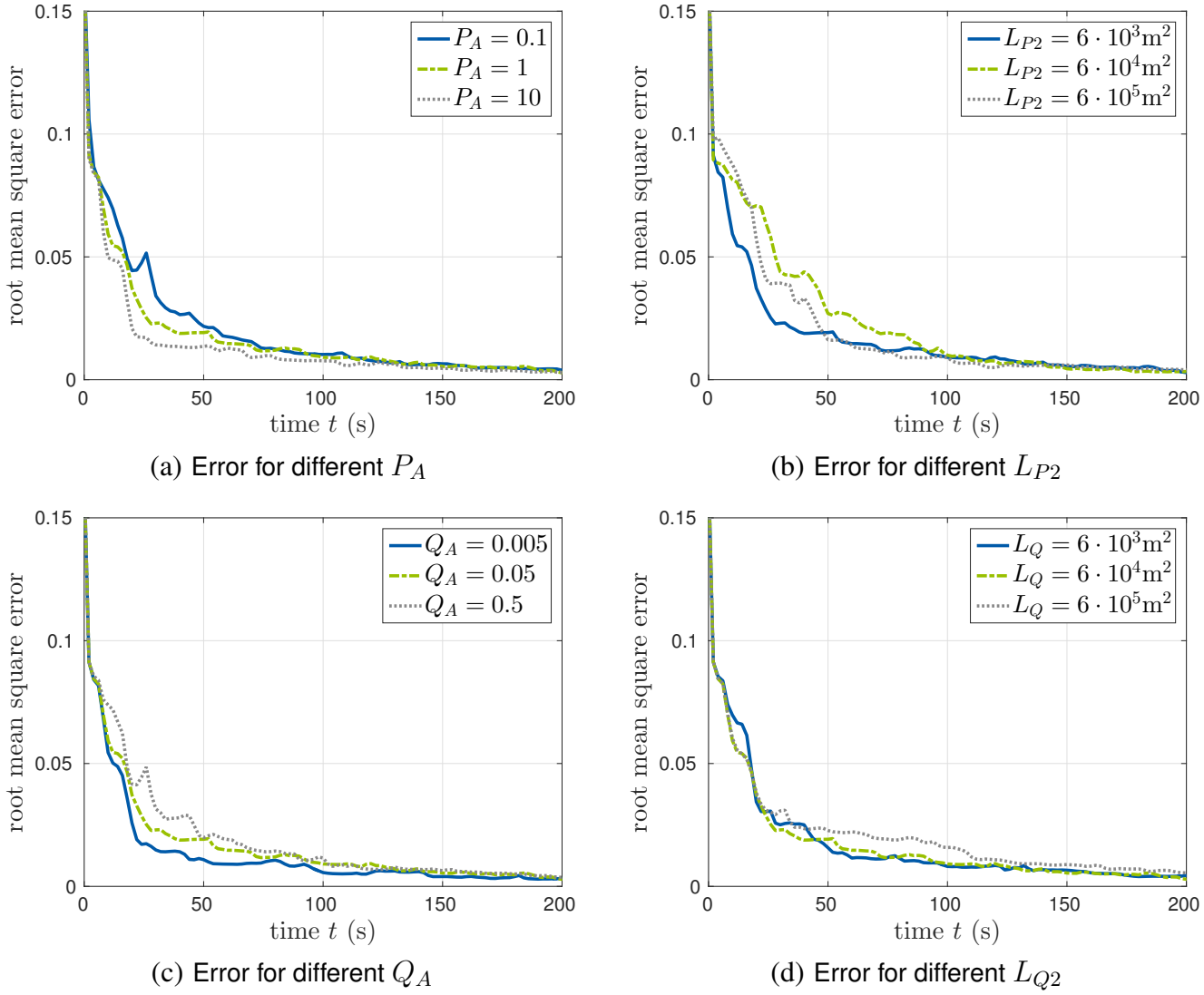
Figure 2.15: Root mean square error of the state estimate obtained with the proposed monitoring approach with KF and EnSRF with different ensemble sizes over time

As expected, the KF provides the best results with the lowest error. However, the results obtained with most of the ensemble approaches is only slightly higher. An ensemble size with 20 ensemble members seems to be most suitable in this context providing comparable results to the Kalman Filter with a lower computational effort. In contrast, an ensemble size of $n_s = 5$ has to be rejected since its error is considerably higher than the one of the other approaches.

Influence of Error Covariance Matrices

Another potentially influencing factor is the choice of the initial error covariance matrix $\mathbf{P}^{(0)}$ and the model error covariance matrix $\mathbf{Q}^{(k)}$. These matrices strongly determine the evolution of $\mathbf{P}^{(k)}$, which is not only important for the data assimilation process (cf. Section 2.4), but is also a decisive factor in the identification of new measurement locations (cf. Section 2.5). In this context, the application of a state-dependent model error covariance matrix (2.52) and (2.54) has been proposed in this work emphasizing positions with higher concentration where a dynamical change is more likely.

The impact of the choice of these matrices is evaluated in a slightly changed test case scenario with initial condition (2.85) and different choices of $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ as well as of $\mathbf{r_1}$ and $\mathbf{r_2}$. Different parameters P_A, P_{L2}, Q_A and Q_{L2} are used to set up the error covariance matrices (2.87) and (2.88).



(a) Error for different P_A

(b) Error for different L_{P2}

(c) Error for different Q_A

(d) Error for different L_{Q2}

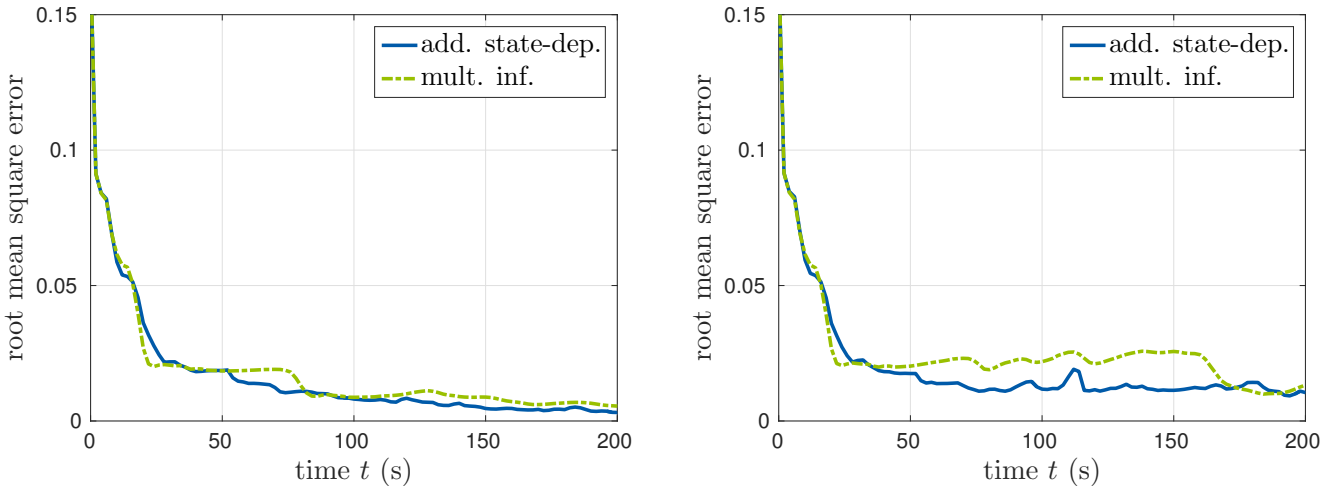
Figure 2.16: Root mean square error of the state estimate obtained with the proposed monitoring approach for different with different initial error covariance matrices $\mathbf{P}^{(0)}$ and model error covariance matrices $\mathbf{Q}$ over time

The obtained results are shown in Figure 2.16. In general, it can be stated that different reasonable choices of the error covariance parameters lead to similar results after a number of time steps. However, if higher quality results should be available at earlier times, some choices are more beneficial than others. The parameter P_A determines the magnitude of the entries in the initial error covariance matrix $\mathbf{P}^{(0)}$. A larger P_A means less trust in the initial estimate. As the initial estimate is rather bad in the considered test case, Figure 2.16(a) recommends the choice of a large P_A in this scenario. On the other hand, if the initial estimate would be rather confident and the measurements are related to larger errors, a small P_A would be the better choice.

Obviously, the opposite applies for the magnitude of Q_A (cf. Figure 2.16(c)). A larger Q_A implies a larger increase of error variance at positions with higher concentration. Starting from an initial condition that is uniformly zero, concentration, and thereby error variance, is introduced to regions in the vicinity of the sensor vehicles. If Q_A is too large, the sensor vehicles might be prevented from globally exploring the domain, resting in regions with higher error variance.

For the parameters L_{P2} and L_Q representing the correlation area, Figure 2.16(b) and 2.16(d) lead to the conclusion that the correlation is spatially strongly limited so that small values of L_{P2} and L_Q provide the best results.

Besides the parameter choices of the error covariance matrix, the performance of the proposed state-dependent model error covariance matrix has to be evaluated. Two different test case settings are run, one with a moderate perturbation (diffusivity perturbed by $0.1\,\mathrm{m^2s^{-1}}$ per step, velocity by $0.5\,\mathrm{ms^{-1}}$ per step with moving average filter) and one with a strong perturbation (diffusivity perturbed by $1\,\mathrm{m^2s^{-1}}$ per step, velocity by $2\,\mathrm{ms^{-1}}$ per step with moving average filter) representing model error. The results obtained with a state-dependent model error covariance matrix (Q_A = 0.05, $L_Q = 6 \cdot 10^4$) are compared to an inflation method ($\alpha = 1.04$), typically used with ensemble filters.



(a) Moderate perturbation

(b) Strong perturbation

Figure 2.17: Root mean square errors of the state estimate obtained with the proposed monitoring approach for state-dependent model error covariance matrix and inflation-based model error covariance matrix for moderate and strong perturbation over time

Figure 2.17 shows the root mean square error of both methods applied as part of the proposed dynamic data-driven monitoring approach. In a scenario with moderate perturbation, the method using state-dependent model error covariance matrices performs already better. However, a significant improvement

using the state-dependent error covariance matrix is obtained when applying it in scenarios with large perturbations. If strong perturbations occur, the growth of error is higher in regions with high concentrations so that the use of state-dependent model error covariance matrices appears to be natural. Such a choice of error covariance matrix is even more suitable when handling a mobile sensor network whose movement is based on the uncertainty of the estimates: vehicles are driven to locations with high concentration estimates where the error is assumed to be highest. On the other hand, if no or hardly model perturbations are added, inflation methods will provide an efficient alternative leading to an exploration of a larger area.

Influence of Number of Sensor Vehicles

A further very important parameter is the number of sensors that are applied in the monitoring approach. Obviously, a higher number of sensor vehicles will provide a higher quality estimate. The conducted test case with different numbers of sensor vehicles confirms this assumption. Figure 2.18 depicts the error for a setting with one, two, three and four sensor vehicles over time. Besides the fact that more sensor vehicles provide better estimates, a saturation behavior can be stated. The benefit of an additional sensor gets smaller with an increasing total number of sensors.

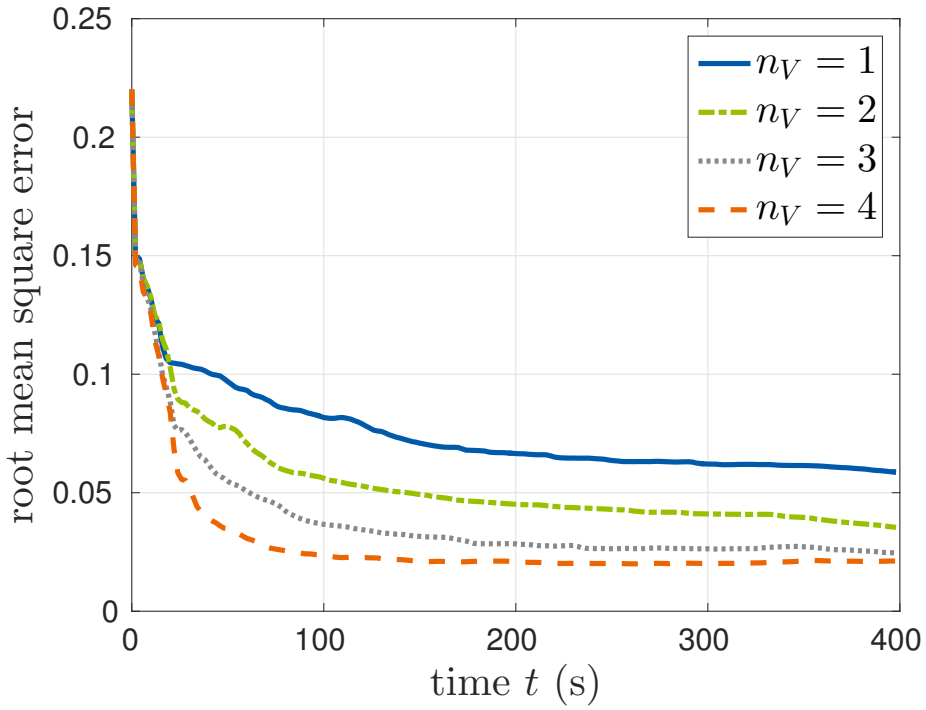


Figure 2.18: Root mean square error of the state estimate obtained with the proposed approach for different numbers $\mathbf{n_V}$ of sensor vehicles over time

Influence of Local Appealing Points

Local appealing points (cf. Subsection 2.5.2) have been designed to pull sensor vehicles into regions with high error variance while they are on their way to the target locations so that uncertainty can be reduced on the way. For the considered test case scenario, the results depicted in Figure 2.19, however, show

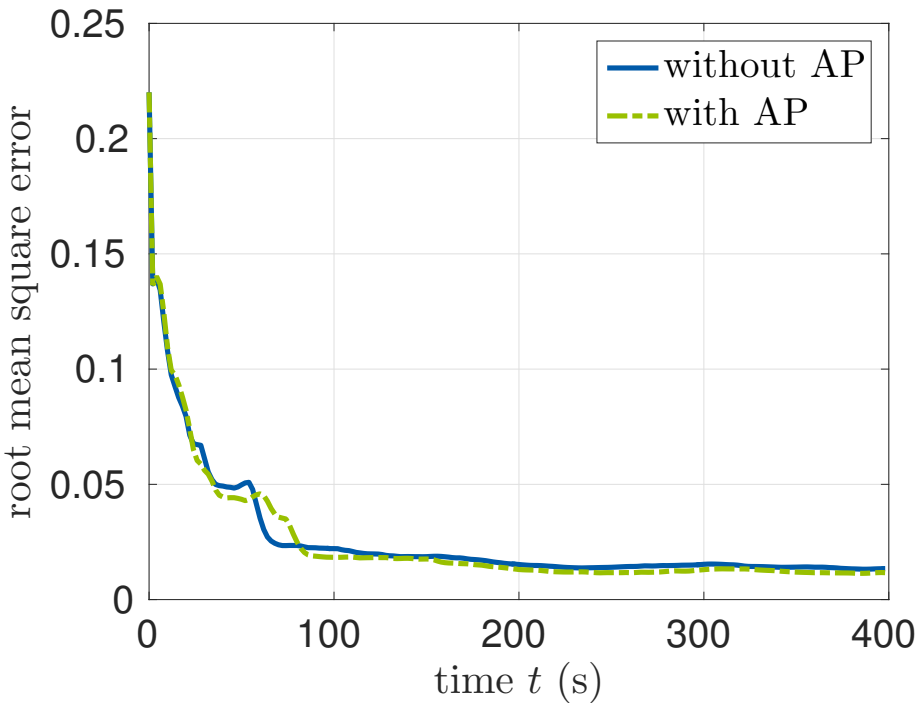


Figure 2.19: Root mean square error of the state estimate obtained with the proposed approach with and without use of local appealing points over time

only a minor improvement for the use of local appealing points. This is mostly caused by the fact that in the considered test case setting most uncertainty is introduced in a relatively dense area. Problems on larger scales with more distributed sources of uncertainty benefit from the use of local appealing points more significantly.

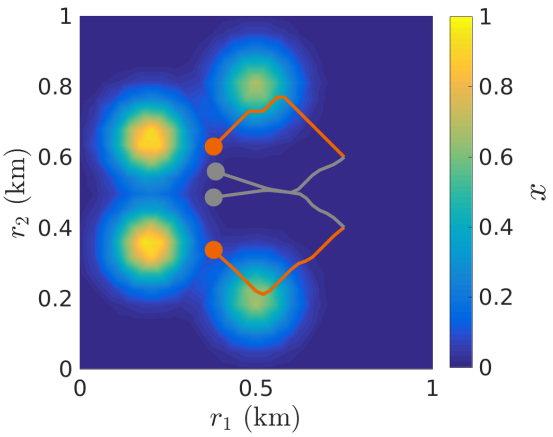
In order to confirm the latter statement, the set up illustrated in Figure 2.20(a) is considered, in which two sensor vehicles are applied to monitor the depicted concentration field with corresponding error covariance matrices. The first target points lie in the centers of the left most Gaussian pulses. While the vehicles without the use of the appealing points move straight to the center, appealing points cause a slight detour so that the error variance in the bottom and top half is reduced on the way. Thereby, the error is decreased significantly (see Figure 2.20(b)).

2.7.4 Results for Joint State and Parameter Estimation

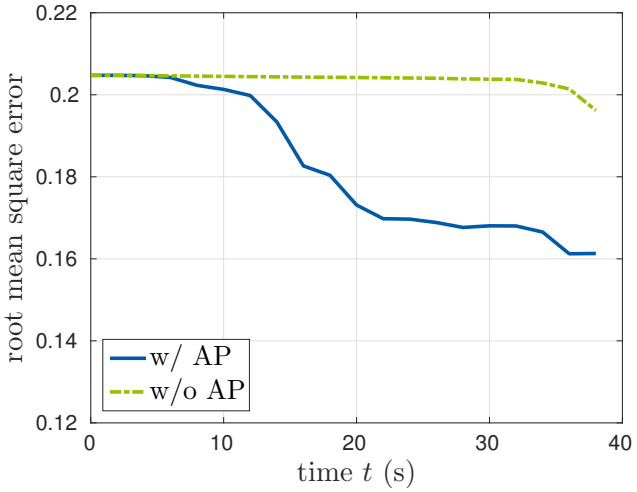
As presented in Subsection 2.4.4, the proposed monitoring approach is also capable of estimating a parameter vector, particularly the source output function $\theta(t)$, along with the state vector. Evaluation of the parameter estimation is the subject of the present subsection and different scenarios with increasing difficulty and complexity are considered.

The general test case setting is adopted from the previous subsections. While velocity and diffusivity remain perturbed, the true as well as the estimated initial condition for the state vector is set to zero $x^{t(0)}(\mathbf{r}) \equiv x^{(0)}(\mathbf{r}) \equiv 0$. Unlike in the previous scenarios, a source is considered. The true source is governed by the equation

$$s(\mathbf{r}, t) = 0.005 f(\mathbf{r} - \mathbf{r_s}, 100\,\mathrm{m}), \tag{2.90}$$



(a) Vehicle trajectories of the approach with appealing points (orange) and withouth appealing points (gray) on true concentration field at initial time



(b) Root mean square error of the monitoring approach with and without appealing points over time

Figure 2.20: Set up, trajectories and results of test case scenario concerning appealing points

Table 2.2: Simulation and controller settings

Forecast covariance magnitude P_A	0.2
Forecast covariance center $\mathbf{r_P}$ [m]	$(250, 250)^T$
Forecast covariance area L_{P1} [m^2]	$1.3 \cdot 10^5$
Forecast covariance area L_{P2} [m^2]	$2 \cdot 10^4$
Model covariance magnitude Q_A	0.02
Model covariance area L_Q [m^2]	$2 \cdot 10^4$
Localization function parameter $P_\theta^{(0)}$	0.02
Localization function parameter Q_θ	0.01

where the function f is a function with local compact support, similar to (4.10) of [63]. The spatial position $\mathbf{r_s}$ and extent of the source are known, but the source output function $\theta^t(t) = 0.005$ is not and has to be estimated jointly with the process state starting from the initial guess $\theta^{(0)} = 0$. Table 2.2 contains the changed settings of the error variance settings involved.

The estimation is processed twice – once using only uncertainty-based target points and once using a source target point instead of an uncertainty-based target point. Figure 2.21 shows the resulting parameter estimates as well as the true solution over time. Both approaches succeed in finding an approximation of the true parameter value. Unfortunately, the estimates suffer from high-frequency fluctuations. These fluctuations stem from the continuous movement of the measurement devices in combination with model error, i.e. perturbed velocity, and diffusivity. A source target point leads to a rest of a sensor vehicle for a longer time and, therefore, at least slightly alleviates the fluctuation effect.

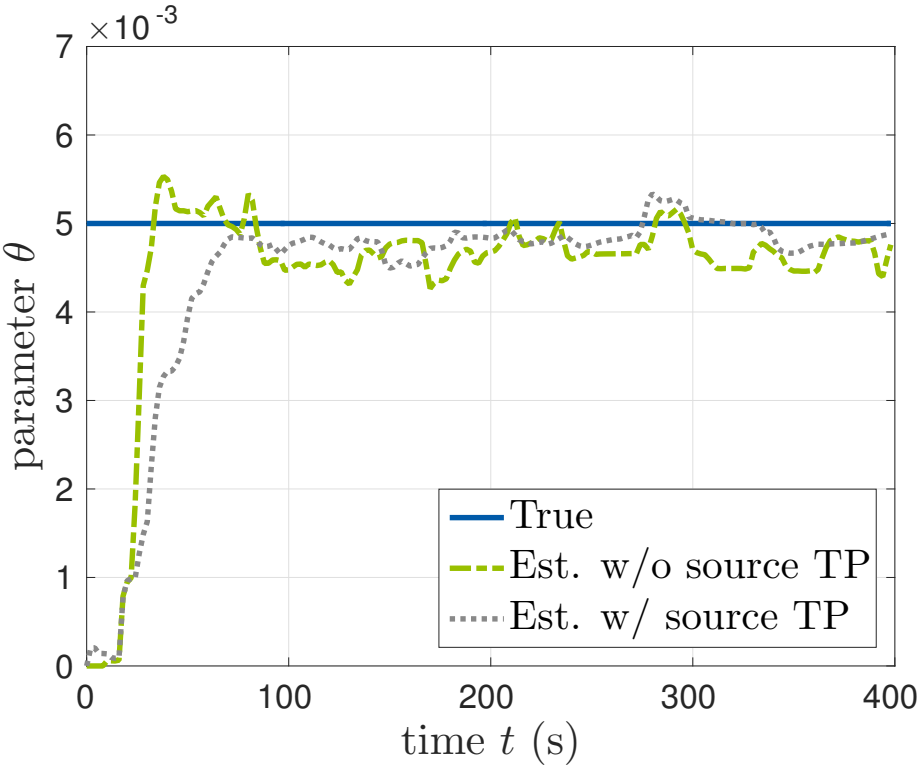


Figure 2.21: Parameter estimate over time for steady parameter scenario

It is not only possible to estimate static parameters, also dynamically changing parameters can be estimated with the parameter estimation method. Therefore, the former test case setting is now considered with a dynamic source

$$s(\mathbf{r}, t) = \left(0.005 + 0.001 \sin\left(\frac{\pi t}{200}\right)\right) f(\mathbf{r} - \mathbf{r_s}, 100). \tag{2.91}$$

The obtained results are shown in Figure 2.22. Apart from some fluctuations, the parameter estimates follow the evolution of the true parameter. Again, a slightly more precise and less fluctuating result is gained if a source target point is applied.

Frequently, the true source position and extent is also not perfectly known and has to be estimated online. Such a scenario can also be solved with the proposed approach considering that the true source function $s^t(\mathbf{r}, t)$ can be approximated by a linear combination of some exemplary source functions $s_i(\mathbf{r}, t)$ whose output functions $\theta_i(t)$ are determined during the online simulation as in the previous test cases. For the considered test case, the true source has the form

$$s^t(\mathbf{r}, t) = 0.005 f(\mathbf{r} - \mathbf{r_s}, 150) \tag{2.92}$$

with $\mathbf{r_s} = (350\,\mathrm{m}, 350\,\mathrm{m})^T$, whereas the considered source functions

$$s_i(\mathbf{r}, t) = \theta_i f(\mathbf{r} - \mathbf{r_{s,i}}, 100) \tag{2.93}$$

have a more narrow width and a different center. The centers of the considered source functions lie along the line $\boldsymbol{\gamma}(\omega) = (700\,\mathrm{m} \cdot \omega, 700\,\mathrm{m} \cdot (1 - \omega))^T$, $\omega \in [0, 1]$: $\mathbf{r}_{\mathbf{s},1} = (330\,\mathrm{m}, 370\,\mathrm{m})^T$, $\mathbf{r}_{\mathbf{s},2} = (380\,\mathrm{m}, 320\,\mathrm{m})^T$, $\mathbf{r}_{\mathbf{s},3} = (430\,\mathrm{m}, 270\,\mathrm{m})^T$.

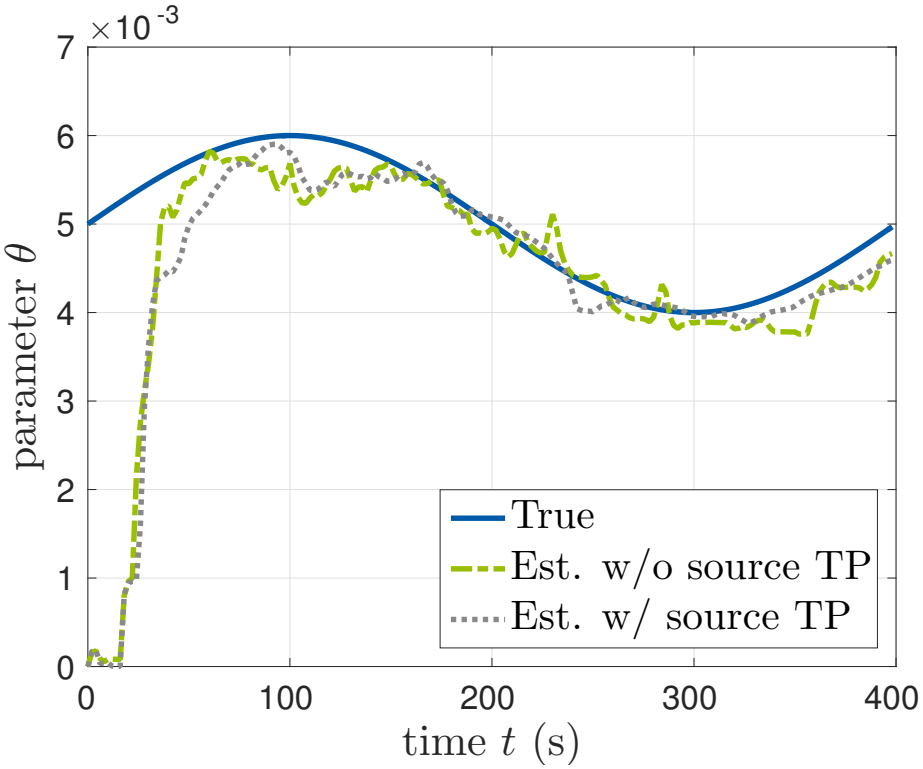


Figure 2.22: Parameter estimate over time for unsteady parameter scenario

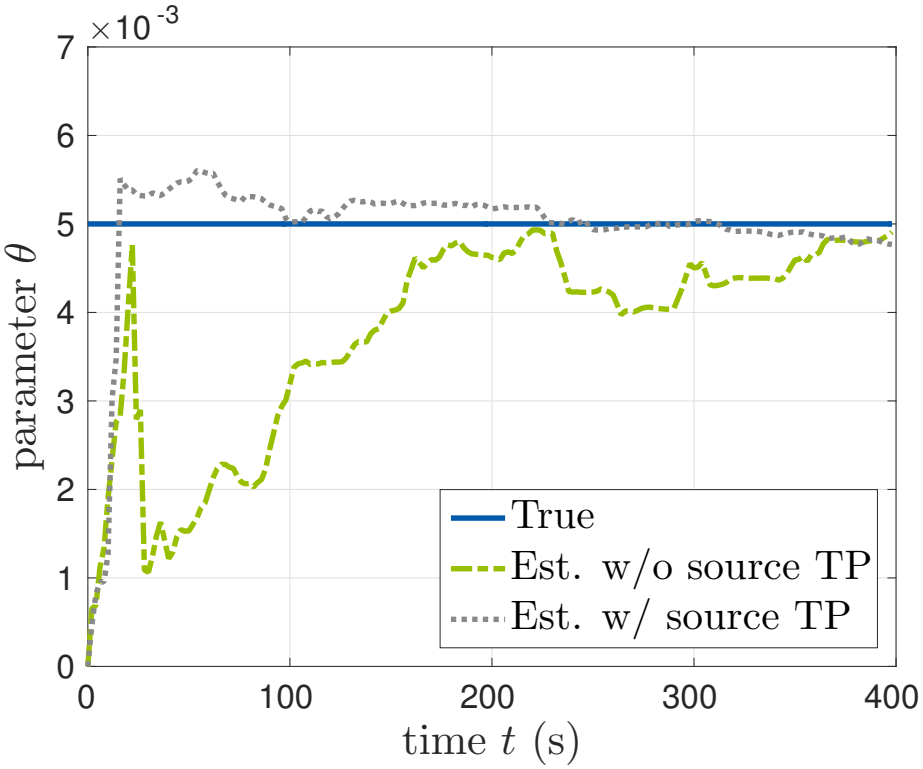


Figure 2.23: Parameter estimate at true source location over time

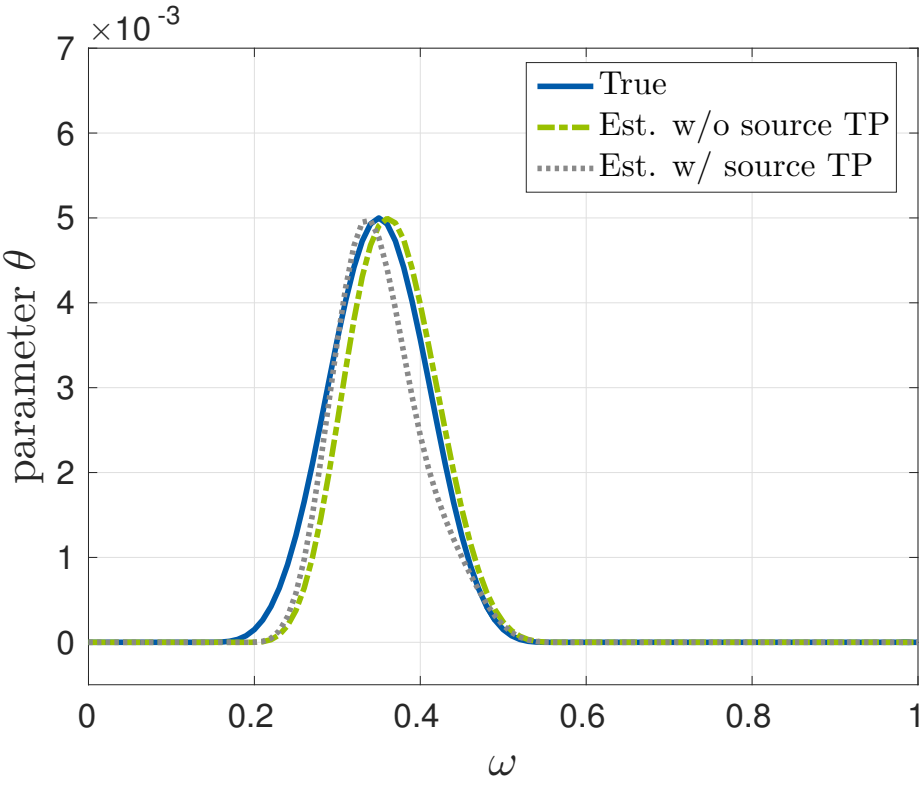


Figure 2.24: Estimated source output along line $\boldsymbol{\gamma}(\omega)$

Starting with the initial estimate $\theta_i^{(0)} = 0$, the results shown in Figure 2.23 are obtained. At the actual source location, the proposed monitoring approach provides a rather precise estimate of the source output parameter, particularly when using a source target point.

Figure 2.24 shows the spatial estimate of the complete source function obtained from linear combining the assumed source functions with their estimates parameters at final time. Considering that only three source functions are available to approximate the true source, the estimate can be stated to be rather good.

2.7.5 Computation Time and Scalability

Besides the estimation quality of the monitoring approach, its computation time is a decisive factor since the approach has to be applied in real-time scenarios in the end. For this reason, the computation time and the scalability of the approach are evaluated in this subsection.

The focus of this work lies in the choice and design of suitable algorithms with a reduced numerical effort, not on the actual optimization of the code for high-performance applications. Therefore, the further examinations are performed with a potentially non-optimized code on a standard machine (Intel (R) Core (TM) i7-3820 3.60 GHz, 16 GB RAM, implemented in Matlab R2015b). A code optimization together with the use of a high-performance computer would further decrease the computation time. However, the following results provide a good first impression of the performance of the single steps and their dependence on certain parameters. In this way, the scalability of the approach with respect to a larger problem dimension, a larger number of sensor vehicles, etc. can be assessed.

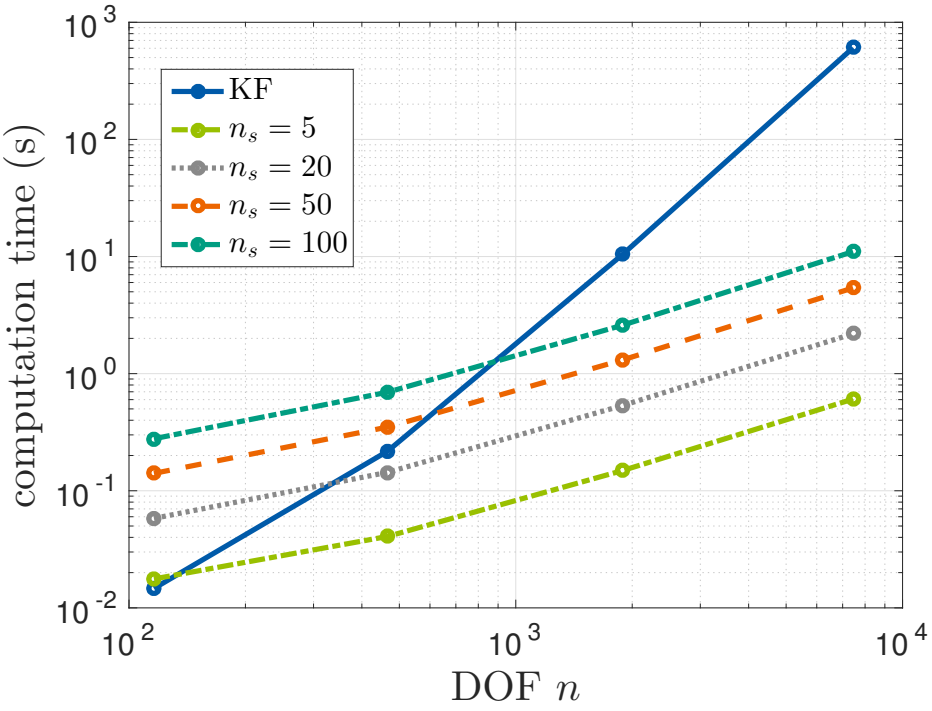


Figure 2.25: Average computation time for one forecast step depending on number of gridpoints n and used data assimilation method with ensemble size n_s

The general test case presented in Subsection 2.7.2 is conducted several times for different settings of n, n_s and n_V and the computation times are averaged. Figure 2.25 provides the required computation times for one forecast step depending on the number of grid points n and the number of ensemble members n_s. As expected, the computing time increases with an increasing number of ensemble members n_S and with an increasing number of grid points n. However, the increase of computation time with increasing n using an EnSRF is much smaller than the increase when using a Kalman Filter. Hence, the EnSRF seems to be a suitable method for application in the considered scenarios. The obtained computation times indicate that the forecast step for a problem of approximately 10,000 grid points can be probably computed in real time in practical scenarios using a central supercomputer. Much higher numbers of grid points could be potentially more problematic.

Figure 2.26 shows the computation times for one analysis step. In general, the effort for the analysis step is lower than the effort of the forecast step, at least for small n_p. Thereby, results can be obtained in a shorter amount of time and the forecast step has to be considered as the critical factor.

The computation times for one control step consisting of target point update and vehicle control are depicted in Figure 2.27. A significant increase of computation time with a growing number of sensor vehicles can be stated. Even for a moderate number of sensor vehicles, the solution of the mixed-integer optimal control becomes time-consuming. The computation time can be decreased by a reduction of the MPC time horizon T_r, a reduction of the number of hyperplanes n_γ or earlier termination of the optimization algorithm. Other strategies to increase scalability with the total number of sensor vehicles are proposed in the following chapters.

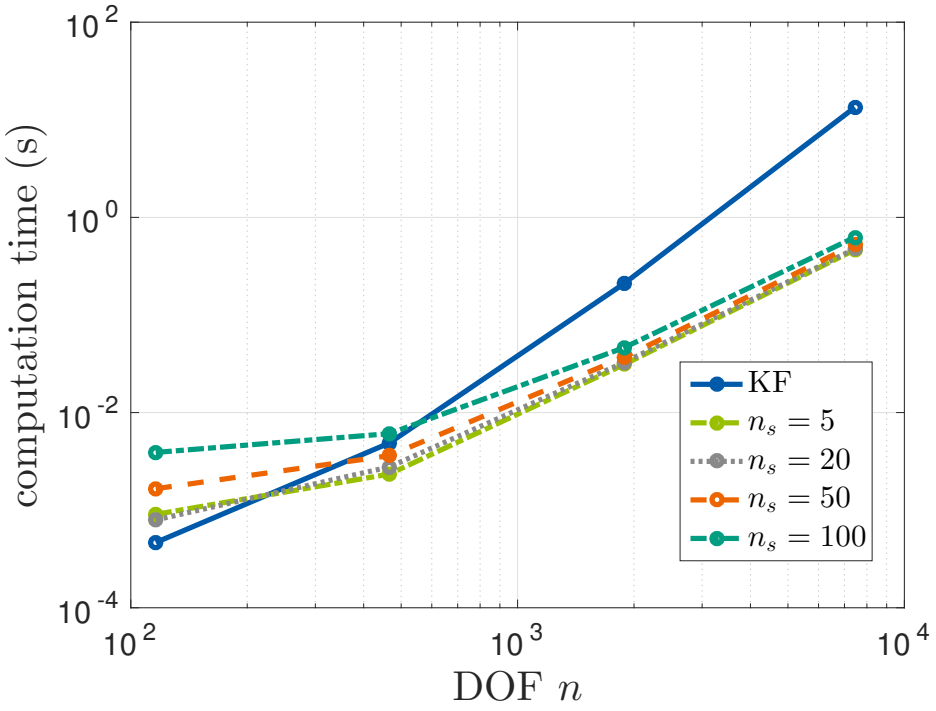


Figure 2.26: Average computation time for one analysis step depending on number of gridpoints n and used data assimilation method with ensemble size n_s

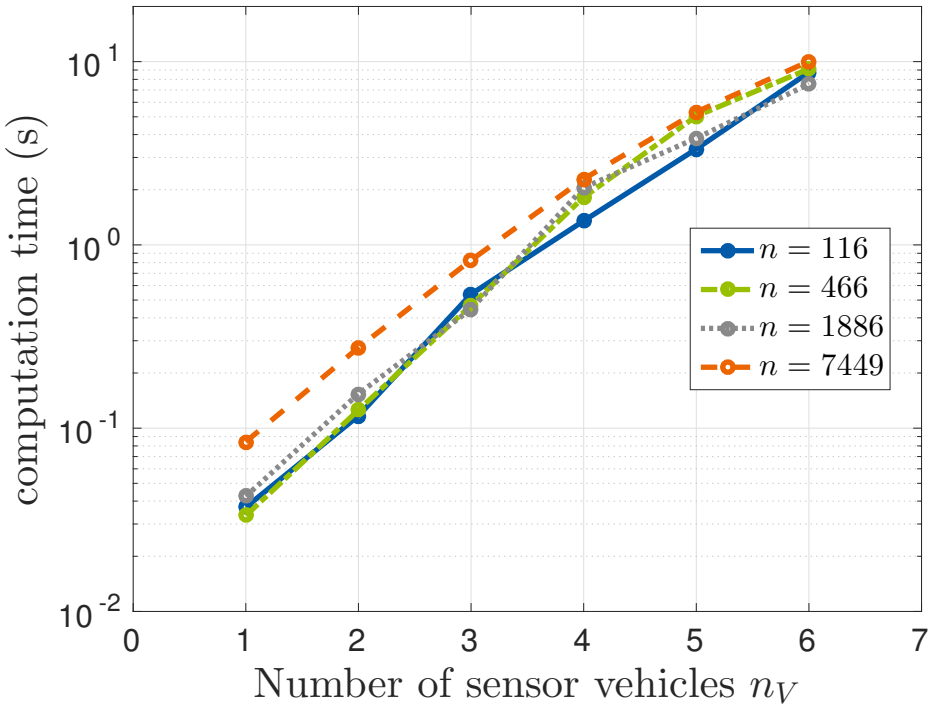


Figure 2.27: Average computation time for one vehicle control step depending on number of sensor vehicles n_V and number of gridpoints n

2.8 Summary and Discussion of Chapter 2

A new dynamic data-driven monitoring strategy for atmospheric dispersion processes has been presented in this chapter. Based on the forecasts of a PDE-model and on observations from a network of multiple mobile sensor vehicles, process state and parameters can be repeatedly estimated in an online fashion and the movement of the sensor vehicles is adapted accordingly. Several building blocks comprising different methods from different research areas have been set up and have been complemented with novel methods (e.g. a sequential target point generation approach) and settings (e.g. state-dependent error covariance matrix) to obtain a DDDAS for atmospheric dispersion monitoring. The resulting structure avoids a sophisticated optimal control problem to generate vehicle control inputs and backs, instead, on a relatively loosely coupled structure of estimation, control, and simulation to account for the online applicability. For the considered test case scenarios, the proposed approach provided good estimation results concerning both process state and source output. The obtained results proved to be by far better than the results obtained with other basic monitoring strategies. Furthermore, the influence of various parameters on the results has been investigated.

However, the proposed approach also comes along with a few drawbacks. When dealing with a lot of sensor vehicles or a very large state dimension, the proposed approach, according to the performed studies of computation time, does not appear to be scalable. The computations might become computationally intractable, even on a supercomputer that is assumed to be applied in the considered scenarios. Dividing the group of sensor vehicles and simplifying the process model could be a solution to this problem. Furthermore, when considering scenarios with characteristic distances of several kilometers, the communication range can become a limiting factor. Sensor vehicles can get out of range and cannot be controlled centrally anymore.

Hence, the proposed approach seems to be suitable for scenarios with a spatial size of several hundred meters providing high communication capabilities with a relatively low number of sensor vehicles and a medium size process model. To account for scenarios for which this centralized approach is not well suited, a closer look is taken on decentralized monitoring approaches in the next two chapters. Decentralized approaches typically do not require vast communication ranges and inherently implement a division of sensor vehicles into smaller groups.

3 A New Decentralized Approach to PDE-based Dynamic Data-Driven Monitoring

This chapter presents a new decentralized dynamic data-driven monitoring approach for atmospheric dispersion processes. Compared to related centralized approaches, the proposed decentralized approach improves scalability and robustness and can be applied in scenarios related to networks with weaker communication capabilities. The proposed approach, unlike related decentralized approaches in literature (cf. Section 1.3), is based on a PDE model of the dispersion process and benefits from its more realistic physical prediction abilities. However, incorporating a greedy PDE model in an application with real-time requirements is highly challenging and simplifications providing a good trade-off between estimate quality and computation time have to be considered. Proper Orthogonal Decomposition, a model order reduction strategy using a low number of basis elements that approximate precalculated solutions of the considered process model, is applied. Furthermore, the aspect of decentrality and limited communication radii have to be taken into account regarding a decentralized data assimilation scheme, a decentralized vehicle controller and a decentralized target point identification method.

While the basis of the work of this chapter has been previously published by the author in [148], an extended variant for snapshot generation and more thorough evaluation results with respect to the accuracy of reduced order models, the influence of decentralized data fusion and the scalability of the approach are presented in the following.

Most of the foundations required for this chapter have already been described in Chapter 2. For an improved understanding, the outline of this chapter adopts the outline of Chapter 2 and the modifications regarding decentrality and further simplifications are explained. Starting with a short review of related decentralized monitoring approaches, an overview of the proposed decentralized monitoring approach is presented and the contributions of the chapter are stated in more detail in Section 3.1. Subsequently, the required building blocks for the monitoring approach are explained in Sections 3.2-3.6. Evaluation results are presented in Section 3.7 and the chapter is concluded with some remarks in Section 3.8.

3.1 A New Decentralized Monitoring Strategy: Contribution and Overview

The goal of this chapter is to design a new decentralized dynamic data-driven monitoring approach for state and parameter estimation. In a decentralized network, a central fusion and control node is completely avoided. Instead, the whole computation is distributed to the sensor vehicles. Each vehicle is equipped with its own computing unit processing only local information regarding data fusion and control. The nodes are capable of communicating within their local neighborhood on a node-to-node basis but have no knowledge of the global communication topology.

The motivation for the application of a decentralized approach is manifold: First, decentralized approaches are especially suitable for larger problem domains that do not fit the communication requirements. As communication only takes part locally, there is no demand for vast communication ranges and the bandwidth requirements are also to a large extent lower. Thereby, scalability with respect to the problem domain improves. Also, scalability with respect to the number of sensor vehicles becomes pos-

sible. Only neighboring sensor vehicles are considered in the computations so that a lot of computation time can be saved. Lastly, also the robustness is increased avoiding a central point of failure.

For these reasons and motivated by the recent hardware developments in the field of sensor vehicles, decentralized mobile sensor networks are increasingly considered for monitoring tasks (cf. the more detailed literature review on this topic in Section 1.3). Most work focuses on distributed coordination [120, 26, 109] of multiple sensor vehicles, i.e. on relatively lightweight tasks for a large number of sensor vehicles. The work on state estimation is relatively sparse. Besides the approach presented in [44], which cannot be considered dynamic data-driven, approaches that are based on relatively simple statistical forecast models have been proposed [170, 90, 62, 38].

Thus, the work presented in this chapter contributes to the state of the art by providing a new decentralized dynamic data-driven monitoring strategy with forecasts based on a PDE model. The proposed approach possesses both, the benefits of DDDAS with its online adaptability to dynamic varying processes and the benefits of PDE-based forecast models, enabling a physically realistic prediction.

However, the use of PDE-based models in dynamic data-driven approaches is highly challenging - even more, when considering decentralized approaches. While the real-time requirements, obviously, still hold, the computational power of the onboard processing units is strictly limited. Maintaining high-dimensional models, which are typical when working with PDE models, is prohibited in this context. Simplifications providing a good trade-off between accuracy and computation time have to be considered.

Another challenge is decentrality. Information is not available globally anymore, but only locally at the sensor vehicles. This requires adapted algorithms for the applied methods. In total, a globally converging strategy has to be designed with locally operating agents.

Like the monitoring approach presented in Chapter 2, the decentralized monitoring approach consists of multiple building blocks. The basis of the building blocks consists of well-established and recently proposed methods from literature as well as of newly developed methods described in the previous chapter. All these methods have to be applied to the present scenario, have to be modified regarding the simplification and decentralization aspect and have to be suitably connected. An overview of the building blocks of the monitoring approach and their interaction is depicted in Figure 3.1.

Every sensor vehicle performs the tasks described in the following locally on its own computing facility.

The cycle is started with the process simulation propagating the mean estimate together with the error covariance to the new measurement time. As this step is very expensive for high-dimensional models, model order reduction techniques are applied to the dispersion model introduced in the previous chapter. Using proper orthogonal decomposition together with a bunch of data representing the characteristic dispersion problem considered in this work drastically reduces the state dimension and, thereby, the computation time. The basics of proper orthogonal decomposition and its specific application to the problem considered within this work, including the proposed choice of the snapshot set, are described in Section 3.2.

Besides the model prediction, the sensor measurement is available as an additional information source. Section 3.3 presents the decentralized observation model and provides more detailed information about the applied communication and computation topology.

The next step consists of data assimilation, i.e. the combination of measurement and model prediction to obtain a higher quality estimate. To account for the decentralized computing architecture with limited communication range, a two-step variant is proposed here. First, every agent integrates its own measurement into the model via the use of the standard Kalman Filter. Second, information concerning the

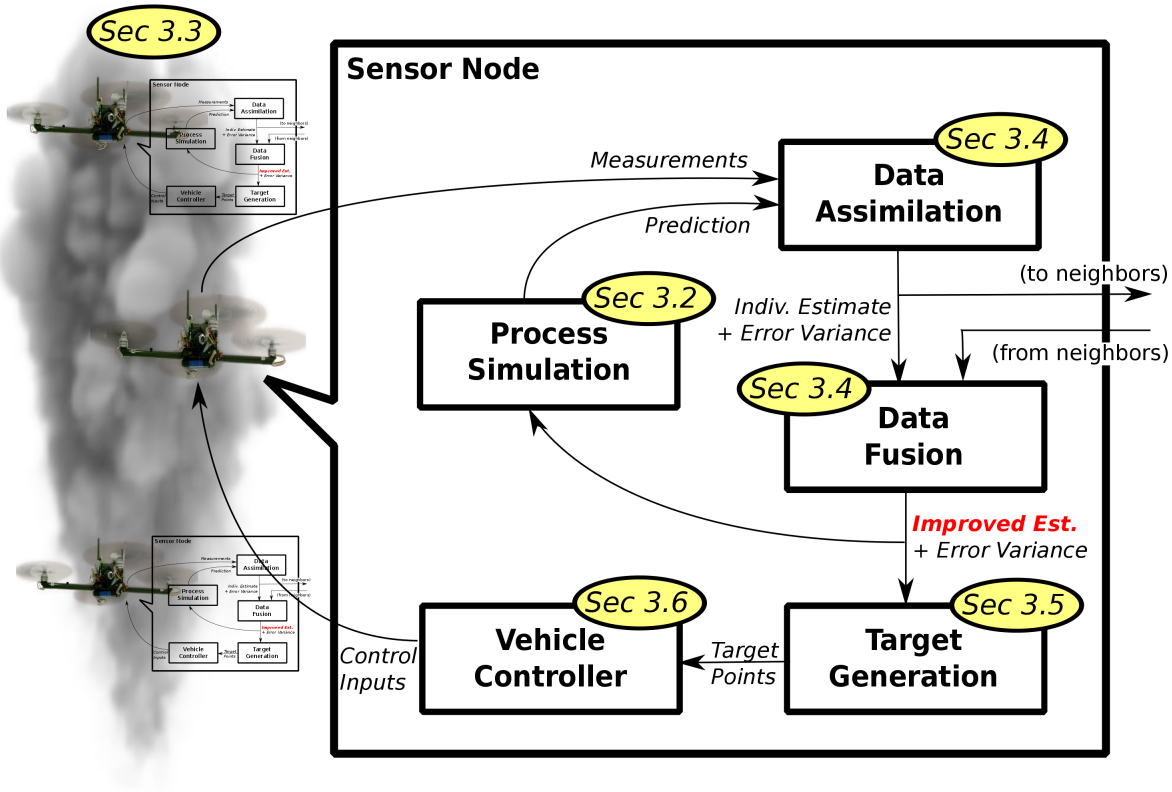


Figure 3.1: Overview of the proposed decentralized monitoring approach

updated state estimates is exchanged with neighbors. Data fusion using Covariance Intersection yields an improved estimate for mean state and corresponding error covariance matrix. The procedure working on reduced models is explained in Section 3.4.

Based on the resulting estimates, suitable measurement locations are generated and updated. For this purpose, the centralized target generation step of the previous chapter is modified for the use in decentralized systems (see Section 3.5).

The generated target points serve as input for the cooperative vehicle controller, whose decentralized variant was already presented in [104] and is applied to the scenario considered in this work. Section 3.6 briefly describes the decentralization of the controller enabling an increased scalability of the approach. The control input for the sensor vehicle is obtained as an output of the controller so that the sensors get to positions that potentially provide more informative measurements.

3.2 Reduced Order Modeling of the Advection-Diffusion Equation

The application of FEM or other spatial discretization methods usually leads to high-dimensional systems since the number of grid points to solve the considered PDE is typically very large. Thus, one has to cope with large state vectors and the forecast equations (2.27) requires the solution of a high-dimensional linear system. A huge computational effort to solve the resulting equations is the consequence. While a central supercomputer is still capable of providing results quickly, at least for a moderate problem size,

these computations become intractable when performing them on the on-board computational units of the sensor vehicles. Thus, the computational burden for model propagation has to be reduced significantly.

One promising approach in this context is to reduce the dimension of the model, i.e. to use a Reduced Order Model (ROM). The idea is to approximate the full dynamics of the system with a lower dimensional system so that the behavior of the process can still be predicted in a reasonable way. Thereby, the ROM can be applied onboard the sensor vehicles for predictions within an appropriate amount of time.

Besides a number of different ROM approaches, especially projection-based approaches are commonly used in practice [10, 15]. In projection-based approaches, the full system is projected to a relatively small number of basis functions. The basis functions to be used depend on the applied model reduction technique.

While the Reduced Basis method [139] is a common method used for parameter-dependent systems, an exemplary projection based approach for state estimation is Balanced Truncation (BT), originally introduced in the field of control design [124]. Applying BT, the system is transformed so that controllability and observability Gramian become diagonal. State variables contributing little to the overall performance are truncated. BT can only be applied to linear systems and affords the solution of a Lyapunov equation, which is expensive for large dimensional systems. To cope with systems with a high number of degrees of freedom and to enable future potential nonlinearity, BT is not a perfect choice in the context of this work.

Instead, the very popular Proper Orthogonal Decomposition (POD) method [115] in its snapshot variant [158] seems to be suitable. Several numerical simulations exemplary for the potential actual system behavior are run prior to the application. Based on the gathered set of information, the POD basis vectors containing characteristics of the solution are constructed.

In this work, Proper Orthogonal Decomposition is used to reduce the dimension of the model. While Subsection 3.2.1 explains the generation of the POD basis, the application of the generated POD basis to the considered problem equations resulting in the desired ROM is treated in Subsection 3.2.2. A specific choice of the snapshot set for problems considered in this work is proposed in Subsection 3.2.3.

3.2.1 Basis Construction via Proper Orthogonal Decomposition

Proper Orthogonal Decomposition, in other areas also known as Principal Component Analysis or Karhunen-Loève expansion, is frequently used for model reduction in a number of different application areas. To compute the POD basis, following the method of snapshots [158], data sampled from several representative simulations or experimental measurements of the system is required. Applying Proper Orthogonal Decomposition to the gathered data, commonly called snapshot set, yields the resulting orthonormal basis $B_{n_\ell} = \{\boldsymbol{\phi}_i\}_{i=1}^{n_\ell}$ with $n_\ell \ll n$.

To set up the snapshot set, several offline computations of the full system are required. Simulation settings, parameters and initial conditions for the different simulation runs should be chosen to be representative for the actual dynamics of the system. Subsection 3.2.3 describes the different simulation settings that are chosen to get a reasonable reduced order model of the state forecast equation (2.27).

Having gathered the solutions $\mathbf{x}_j^{(k)}, k \in \{0, \dots, n_T\}, j \in \{1, \dots, n_S\}$ of the n_S different simulation runs at time $t^{(k)}$, the snapshot matrix

$$\mathbf{C} = (\mathbf{x}_1^{(0)}, \dots, \mathbf{x}_1^{(n_T)}, \mathbf{x}_2^{(0)}, \dots, \mathbf{x}_{n_S-1}^{(n_T)}, \mathbf{x}_{n_S}^{(0)}, \dots, \mathbf{x}_{n_S}^{(n_T)}) \in \mathbb{R}^{n \times n_C} \tag{3.1}$$

with $n_C = n_S(n_T + 1) < n$ can be constructed. Note that the snapshot matrix size n_C, and with it the effort for basis construction, can be reduced if the solution is not inserted into the snapshot set for every time $t^{(k)}$, but only at specific times.

The aim is now to find an orthonormal basis of size n_ℓ such that the error between the snapshots represented in the full space and the snapshots projected onto the reduced space is minimized. To attain this, the following optimization problem has to be solved

$$\begin{aligned} \min_{B_\ell} \quad & \sum_{j=1}^{n_C} \left\| (\mathbf{C})_{\cdot,j} - \sum_{i=1}^{n_\ell} \left(\left((\mathbf{C})_{\cdot,j} \right)^T \boldsymbol{\phi}_i \right) \boldsymbol{\phi}_i \right\|^2 \\ \text{s.t.} \quad & \boldsymbol{\phi}_i^T \boldsymbol{\phi}_j = \delta_{ij}, \end{aligned} \tag{3.2}$$

where δ_{ij} denotes the Kronecker delta. It turns out (cf. [77]) that the solution of the constrained optimization problem (3.2) can be equivalently found by solving the symmetric $n_C \times n_C$ eigenvalue problem

$$\mathbf{C}^T \mathbf{C} \mathbf{v}_i = \lambda_i \mathbf{v}_i \tag{3.3}$$

and generating the POD basis B_{n_ℓ} from

$$\boldsymbol{\phi}_i = \frac{1}{\sqrt{\lambda_i}} \mathbf{C} \mathbf{v}_i \tag{3.4}$$

for $i \in \{1 \dots n_\ell\}$, where λ_i are sorted in a descending order $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{n_C} \geq 0$. Eigenvectors $\mathbf{v}_i$ corresponding to large eigenvalues λ_i contribute most to the representation of the solution given by the snapshot set, whereas the influence of eigenvectors corresponding to small eigenvalues is minor. Motivated by this, eigenvectors with smaller eigenvalues are neglected and a reduced basis B_{n_ℓ} with only n_ℓ basis elements is obtained.

In general, the number n_ℓ of POD basis elements should be chosen to be as small as possible. The fraction δ to which the chosen basis captures the total energy of the full snapshot set can be determined by the relation

$$\frac{\sum_{i=1}^{n_\ell} \lambda_i}{\sum_{i=1}^{n_C} \lambda_i} = \delta. \tag{3.5}$$

Thus, for a given nominal value δ_S, the number n_ℓ should be chosen to be the minimal number such that $\delta > \delta_S$.

3.2.2 POD-based Reduced Order Model

Having computed the POD basis, the full forecast system (2.33) can be reformulated in a lower dimensional space by projecting the corresponding vectors and matrices on the POD basis. With the POD matrix $\boldsymbol{\Phi} = \text{row}\{\boldsymbol{\phi}_i\}_{i=1}^{n_\ell} \in \mathbb{R}^{n \times n_\ell}$, the reduced state vector $\hat{\mathbf{x}}$ can be expressed as $\hat{\mathbf{x}}^{(k)} = \boldsymbol{\Phi}^T \mathbf{x}^{(k)}$ and the reduced variant of (2.33) yields

$$\hat{\mathbf{x}}^{(k+1)} = \hat{\mathbf{M}} \hat{\mathbf{x}}^{(k)} + \hat{\mathbf{f}}^{(k+1)}, \tag{3.6}$$

with $\hat{\mathbf{M}} = (\boldsymbol{\Phi}^T (\mathbf{W_1} \mathbf{V_1} + \mathbf{W_2}) \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \mathbf{W_1} \mathbf{V_2} \boldsymbol{\Phi}$ and $\hat{\mathbf{f}}^{(k)} = \boldsymbol{\Phi}^T \mathbf{f}^{(k)}$. With this reduced forecast model, state operations like the state forecast become much more efficient. This is even more significant for the propagation of the reduced error covariance matrix $\hat{\mathbf{P}}^{(k)} = \boldsymbol{\Phi}^T \mathbf{P}^{(k)} \boldsymbol{\Phi}$, which, according to (2.35), reads

$$\hat{\mathbf{P}}^{(k+1)} = \hat{\mathbf{M}} \hat{\mathbf{P}}^{(k)} \hat{\mathbf{M}}^T + \hat{\mathbf{Q}}^{(k+1)} \tag{3.7}$$

with $\hat{\mathbf{Q}}^{(k)} = \boldsymbol{\Phi}^T \mathbf{Q}^{(k)} \boldsymbol{\Phi}$.

3.2.3 Snapshot Generation

Several simulation runs have to be conducted prior to the application of the reduced model to build up the snapshot set. As the quality of the results obtained with the reduced model strongly depends on the representativeness of the snapshot set, the choice of the snapshot set should be well considered. Different factors and parameters that potentially influence the solution have to be identified and must be represented in the simulation runs building up the snapshot set.

In the context of this work, the true state evolution (2.34) with explicit consideration of the source yields

$$\mathbf{x}^{\mathbf{t}(k+1)} = \mathbf{M}\mathbf{x}^{\mathbf{t}(k)} + \mathbf{S}\boldsymbol{\theta}^{\mathbf{t}(k+1)} + \mathbf{f}^{(k+1)} + \boldsymbol{\eta}^{(k+1)} \tag{3.8}$$

and has to be represented by the snapshot set. As the true initial condition, the source terms and model errors are not exactly known, simulations are performed with initial guesses of these entities and potential variations of them.

To account for different initial conditions, several simulation runs have to be performed based on different initial conditions for the concentration. Assuming a set of n_{IC} different initial conditions $x_j^{(0)}(\mathbf{r}) = q_{IC,j}(\mathbf{r})$ with $j \in \{1, \ldots, n_{IC}\}$, the solution of a linear model with initial conditions consisting of a linear combination of the individual conditions

$$x^{(0)}(\mathbf{r}) = \sum_{j=1}^{n_{IC}} \alpha_j q_{IC,j}(\mathbf{r}) \tag{3.9}$$

is also represented by the snapshot set. This motivates the use of several basis functions for $q_{IC,j}(\mathbf{r})$ enabling the approximation of a large set of different potential initial conditions.

For the source term $s(\mathbf{r}, t) = \theta(t)q(\mathbf{r})$, impulse response snapshots, as for example applied in [67] for the control of fluid equations, can be generated. To represent the source effect, it is sufficient to conduct several simulations of the following system corresponding to different source location functions $q_j(\mathbf{r})$, $j \in \{1, \ldots, n_{SS}\}$ with discretized vector $\mathbf{S}_j$ as instantaneous source

$$\mathbf{x}_j^{\mathbf{S}(k+1)} = \mathbf{M}\mathbf{x}_j^{\mathbf{S}(k)} + \mathbf{f}^{(k+1)} + \delta_{1(k+1)}\mathbf{S}_j \tag{3.10a}$$

$$\mathbf{x}_j^{\mathbf{S}(0)} = \mathbf{0}. \tag{3.10b}$$

According to the use of the Kronecker delta $\delta_{1(k+1)}$, the source term $\mathbf{S}_j$ only has to be considered in the first time step. The obtained snapshots capture the effect of all possible linear combinations $\sum_{j=1}^{n_{SS}} \theta_j^{(k)}\mathbf{S}_j$, even for unsteady $\theta_j^{(k)}$. The reason why it is sufficient to calculate the impulse response $\mathbf{x}_j^{\mathbf{S}(k)}$ as in system (3.10) is that system (3.8) can be regarded as LTI system with inputs $\mathbf{S}\boldsymbol{\theta}^{\mathbf{t}(k)}$ so that, according to Duhamel's principle, it is possible to express the solution $\mathbf{x}^{(k)}$ of the system as the sum of the convolution of the parameter θ_j with the impulse response $\mathbf{w}_j$ corresponding to an initial excitation $\mathbf{S}_j^{(0)}$

$$\mathbf{x}^{(k)} = \sum_{j=1}^{n_\theta} \sum_{i=0}^{k} \theta_j^{(i)} \mathbf{x}_j^{\mathbf{S}(k-i)}. \tag{3.11}$$

Finally, the snapshot set is enriched with the influence of the assumed model error to be able to continuously represent the evolution of the true state. The model error $\boldsymbol{\eta}$ can also be considered to be an

input of the LTI system so that the impulse response technique can be applied for the model error again. Therefore, the Karhunen-Loève expansion of the model error

$$\boldsymbol{\eta}^{(k)} = \sum_{j=1}^{n} \sigma_j^{(k)} \sqrt{\lambda_j} \mathbf{v}_j \tag{3.12}$$

is used, where λ_j and $\mathbf{v}_j$ are eigenvalues and eigenvectors of the error covariance matrix $\mathbf{Q}$ and $\sigma_j^{(k)}$ is a normally distributed random variable. An approximation to the noise term is obtained if the summation in (3.12) is only made over the n_η most dominant eigenvectors. If this ansatz is plugged into (3.8), the noise can be interpreted as random input and Duhamel's principle can be applied again. The impulse responses $\mathbf{x}_j^{\mathbf{S}\boldsymbol{\eta}}$ for all considered noise eigenvectors $\mathbf{v}_j$ are added to the snapshot set.

Please note that the error covariance matrices considered in this work are typically assumed to be time- or state-dependent so that, due to different $\mathbf{Q}$ a high number of different model error snapshot simulations would have to be performed. Moreover, the true model error to be represented is commonly not normally distributed (see Section 2.4.1) and the error covariance matrix is not known exactly. While a general choice of $\mathbf{Q}$ can still lead to some representative snapshot ingredients, it is also advisable to consider also other potential perturbations effects for integration into the snapshot set.

Due to linearity of (3.8), all three kinds of snapshot simulations can be performed individually without the need of consideration of the effects of the other terms.

3.3 Modeling of Sensor Vehicles

The monitoring approach considered in this chapter is decentralized and limited communication ranges are considered. Thus, compared to Chapter 2, some modifications on the sensor vehicle models have to be made. While the dynamics are the same as described in Subsection 2.3.1, a new communication and computation topology has to be designed (see Subsection 3.3.1) and the observation model has to be adapted (see Subsection 3.3.2).

3.3.1 Decentralized Computing Architecture

The computation architecture considered in this chapter is fully decentralized. This means that there is no central computing node. Instead, every vehicle is equipped with a computing facility, on which local computations can be performed. Communication is only possible with neighboring nodes. The benefits of such an architecture are the avoidance of a central point of failure, a potentially more reactive sensor fleet, and an increased scalability. Furthermore, restricted communication capabilities are explicitly taken into account.

In this section, it is assumed that communication is only possible with vehicles residing within the limited communication range d_j. Defining for sensor vehicle $j \in \{1, \ldots, n_V\}$ the set of neighbors $N_j(t) = \{m \in \{1, ..., n_V\} : \|\mathbf{r}_j^{\mathbf{p}}(t) - \mathbf{r}_m^{\mathbf{p}}(t)\| < d_j\}$, vehicle j is capable to communicate with all vehicles in N_j. Due to vehicle movement, the structure is not static, but changes over time. Figure 3.2 shows an exemplary communication structure with point to point communication between any two connected sensor vehicles.

Applying the proposed structure, every sensor vehicle maintains its own model state, which is propagated in time on the local computing device and updated with information from neighboring sensor vehicles.

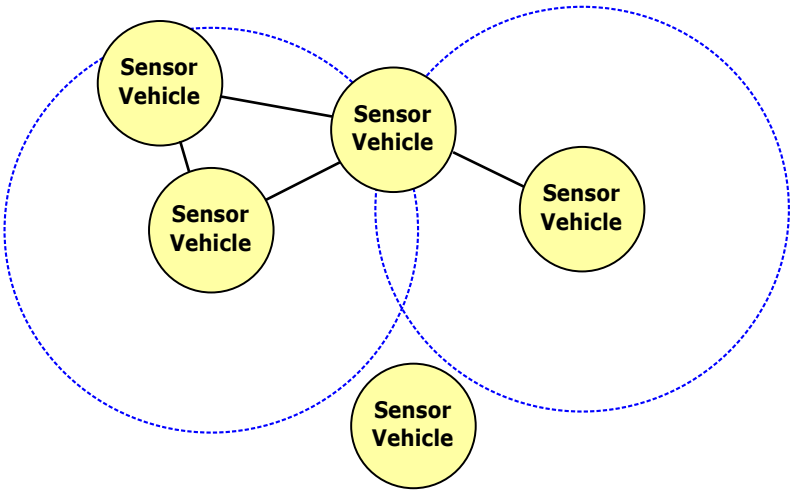


Figure 3.2: Communication network of the proposed decentralized monitoring approach. Each vehicle communicates with vehicles located within the limited communication range (depicted in blue). For clarity, only two communication ranges are shown.

Model states of different sensor vehicles can differ so that information concerning the whole model state is exchanged in the vicinity.

3.3.2 Observation Model

Compared to the centralized observation model described in Subsection 2.3.2, a minor modification is needed to set up the decentralized observation model for every sensor vehicle j.

In general, vehicle j does not have global knowledge of the complete system, i.e. $N_j \subseteq \{1, \ldots, n_V\}$ and (2.32) is unknown. Only the observation model

$$\mathbf{y}_j^{(k)} = \mathbf{H}_j^{(k)}(\mathbf{x}^{\mathbf{t}(k)}) + \boldsymbol{\epsilon}^{(k)}. \tag{3.13}$$

with observation vector $\mathbf{y}_j = \text{col}\{y_i : i \in N_j\}$, observation model operator $\mathbf{H}_j = \text{col}\{\mathbf{h}_i : i \in N_j\}$, model error $\boldsymbol{\epsilon}_j = \text{col}\{\epsilon_i : i \in N_j\}$ and corresponding model error covariance matrix $\mathbf{R}_j = \text{blockdiag}\{R_i : i \in N_j\}$ could be accessed. However, for the decentralized data fusion method described in the following section, it is even sufficient to consider only the own measurement y_j according to (2.30).

3.4 Decentralized Data Fusion

Instead of fusing data at a central point of the system, the data assimilation procedure has to take place locally at every sensor vehicle in a decentralized architecture. Every node updates its own model state and error statistics with its own measurements as well as with information obtained from the neighbors. Thus, the centralized data assimilation step presented in Subsection 2.4 has to be modified to fit the decentralized architecture and methods from distributed data fusion [126, 29] have to be applied.

A simple solution for decentralized data assimilation would be to send all measurements to all nodes, to assemble the global observation vector $\mathbf{y}$ and to apply the standard Kalman update. To avoid waiting

for all measurements to arrive at the node, the Kalman formulation is reformulated into its information form in practice. Not observations concerning the state, but observation information corresponding to an information measure for the state estimate are communicated. As fusion of information is additive, the fusion process becomes associative and newly obtained information can be assimilated immediately.

With increasing number of sensors, such a Kalman or Information Filter approach becomes impractical since the information from one sensor has to be sent to every other sensor vehicle resulting in a quadratically increase in communication effort. More scalable approaches, therefore, are restricted to a communication only within the neighborhood. In these approaches, the measurements are commonly only assimilated locally. However, this results in different state estimates at different sensor vehicles. To generate a consistent global state estimate, estimates have to be exchanged between neighboring nodes and have to be combined with the own estimate. Examples for such approaches are diffusion-based approaches [32] or consensus approaches [131]. The drawback of these approaches is that the error covariance matrix is not involved in finding the fused state estimate so that the updated estimate might be worse than the prior one.

For dynamical systems as considered within the scope of this work, the approaches presented above cannot be applied. Every sensor vehicle can be assumed to be an expert on the part of the domain it is currently located so that the underlying statistics should be considered in the fusion process. Furthermore, due to the movement of the sensor vehicles, it is not guaranteed that the network is fully connected all the time leading to an even larger deviation of the individual state estimates. The question is, thus, how the state estimates of two neighboring nodes can be fused regarding the underlying uncertainty. It has to be obeyed that state vectors at two different sensor vehicles can be highly correlated since they use common models, might have already fused common state and measurement information, and rely on further common prior information. As the correlation is usually unknown, application of Kalman Filter techniques is prohibited in this context. In this work, Covariance Intersection [91] providing a consistently fused estimate by assuming a worst-case scenario considering the analysis error covariance matrix is applied to resolve this situation.

In the following, a two-step data fusion method is proposed for the application scenarios considered in this work. First, the own measurement is combined with the present model forecast using the Kalman Filter (see Subsection 3.4.1). Then, information based on the state update is exchanged with neighboring units and the information is fused using CI (see Subsection 3.4.2). With the use of a reduced order model (cf. Subsection 3.2), the size of the exchanged information suits the communication capabilities and the application of KF and CI, more greedy fusion methods, is justified.

3.4.1 Kalman Filter

Every time a new measurement gets available, it is combined with the individual state forecast present at the corresponding sensor vehicle using the Kalman Filter described in Subsection 2.4.2. The Kalman Filter is applied according to the description in the previous chapter, including the forecast and analysis stage with the difference that only the own measurement (3.13) is incorporated. Thus, the Kalman update

scheme for vehicle $j \in \{1, \ldots, n_V\}$ reads as follows (time indices are omitted in the following since all entities refer to the same time stage):

$$\hat{\mathbf{k}}_j = \hat{\mathbf{P}}_j^{\mathbf{f}} \hat{\mathbf{h}}_j^T \left(\hat{\mathbf{h}}_j \hat{\mathbf{P}}_j^{\mathbf{f}} \hat{\mathbf{h}}_j^T + R_j \right)^{-1} \tag{3.14a}$$

$$\hat{\mathbf{x}}_j^{\mathbf{a},-} = \hat{\mathbf{x}}_j^{\mathbf{f}} + \mathbf{k}_j \left(y_j - \hat{\mathbf{h}}_j \hat{\mathbf{x}}_j^{\mathbf{f}} \right) \tag{3.14b}$$

$$\hat{\mathbf{P}}_j^{\mathbf{a},-} = \left(\mathbb{1} - \hat{\mathbf{k}}_j \hat{\mathbf{h}}_j \right) \hat{\mathbf{P}}_j^{\mathbf{f}}, \tag{3.14c}$$

where the measurement y_j and the observation error covariance R_j are scalars, the reduced Kalman gain $\hat{\mathbf{k}}_j \in \mathbb{R}^{n_\ell}$ is a column vector and the reduced observation operator $\hat{\mathbf{h}}_j \in \mathbb{R}^{n_\ell}$ is a row vector. However, (3.14) comprises only the first part of the analysis so that the Kalman updated entities are denoted by the superscript $(\cdot)^{a,-}$.

3.4.2 Covariance Intersection

Having updated the own state vector with the own measurement, the next step comprises the fusion of the own state vector with information of neighboring state vectors.

Therefore, the state vector as well as the error covariance matrix has to be exchanged with the neighbors. Instead of exchanging $\hat{\mathbf{x}}_j^{\mathbf{a},-}$ and $\hat{\mathbf{P}}_j^{\mathbf{a},-}$ directly, the corresponding information vector $\hat{\mathbf{i}}_j^{\mathbf{a},-} \in \mathbb{R}^{n_\ell}$ and information matrix $\hat{\mathbf{I}}_j^{\mathbf{a},-} \in \mathbb{R}^{n_\ell \times n_\ell}$ with

$$\hat{\mathbf{i}}_j^{\mathbf{a},-} = \left(\hat{\mathbf{P}}_j^{\mathbf{a},-} \right)^{-1} \hat{\mathbf{x}}_j^{\mathbf{a},-} \tag{3.15}$$

$$\hat{\mathbf{I}}_j^{\mathbf{a},-} = \left(\hat{\mathbf{P}}_j^{\mathbf{a},-} \right)^{-1} \tag{3.16}$$

are exchanged. The use of the information metrics saves computational time for the further fusion step as the matrix inversion (3.16) only has to be performed once per node.

Unfortunately, neighboring state vectors might be highly correlated relying on common updates in the past so that a standard Kalman method cannot be used for fusion. This would only be possible if the correlation is explicitly taken into account, but the maintenance of additional covariance matrices would be much too expensive in this context.

An alternative is provided by Covariance Intersection (CI) [91, 93]. Using CI, the correlation between the vectors to be fused do not have to be known, whereas the error covariance matrices influence the fusion results. The result is a conservative, but consistent fused mean state vector with corresponding error covariance matrix. The latter presents an upper bound for the actual error variance that would have resulted from a fusion considering correlation.

CI uses a convex combination of state vectors and error covariances to generate the analysis state vector $\hat{\mathbf{x}}_j^{\mathbf{a},+}$ and error covariance matrix $\hat{\mathbf{P}}_j^{\mathbf{a},+}$

$$\hat{\mathbf{P}}_j^{\mathbf{a},+} = \left(\sum_{m \in N_j} \omega_{j,m} \hat{\mathbf{I}}_m^{\mathbf{a},-} \right)^{-1} \tag{3.17}$$

$$\hat{\mathbf{x}}_j^{\mathbf{a},+} = \hat{\mathbf{P}}_j^{\mathbf{a},+} \left(\sum_{m \in N_j} \omega_{j,m} \hat{\mathbf{x}}_m^{\mathbf{a},-} \right). \tag{3.18}$$

The parameters $\omega_{j,m}$ can be found by minimizing the trace of the resulting covariance matrix

$$\begin{aligned} \min_{\omega_{j,m}} \quad & \text{trace}\left(\left(\sum_{m\in N_j,} \omega_{j,m}\hat{\mathbf{I}}_m^{\mathbf{a},-}\right)^{-1}\right) \\ \text{s.t.} \quad & \omega_{j,m} \geq 0, \sum_{m\in N_j} \omega_{j,m} = 1. \end{aligned} \tag{3.19}$$

This means that a nonlinear optimization problem (3.19) has to be solved to obtain a consistent fusion result. For the fusion of a relatively high number of state vectors, this can become a severe drawback from a computational point of view. Sequential CI approaches have been proposed counteracting this problem by sequentially solving a one-dimensional optimization problem [47].

3.5 Decentralized Identification of Measurement Locations

The decentralized identification of suitable target points follows the centralized determination as described in Section 2.5. Based on the error covariance matrix and estimated source locations, a set of target points is sequentially identified and updated from time to time.

Compared to the explanations in Section 2.5 the following remarks have to be obeyed when applying a decentralized architecture.

- *Every node identifies own set of target point:* There is no global set of target points, but every vehicle unit defines its individual set of target points using the strategies described in Section 2.5. As the state and the corresponding error covariance at two different sensor vehicles are potentially different, the target points at both sensor vehicles are probably also not identical. While the sets at two sensor nodes far away from each other probably deviate to a large degree, neighboring sets of target points are relatively similar as they have exchanged state information in the previous step so that they still can cooperate with respect to target allocation.
- *Error variance of full system is needed:* When using reduced order models, the error covariance matrix of the full system $\mathbf{P}_j$, on which Algorithm 1 is based is not available. However, the reduced error covariance matrix $\hat{\mathbf{P}}_j$ allows no conclusions about interesting measurement locations as its basis has global support. Thus, the full system's error covariance matrix has to be examined. To avoid the complete reconstruction of the full error covariance matrix, a simplified variant can be applied that is sufficient to determine the error variance $\mathbf{v}_j = \text{diag}(\mathbf{P}_j)$

$$\boldsymbol{\Xi}_j = \boldsymbol{\Phi}\hat{\mathbf{P}}_j \tag{3.20}$$

$$\left(\mathbf{v}_j\right)_i = \sum_{l=1}^{n_\ell} \left(\boldsymbol{\Xi}_j\right)_{i,l} \left(\boldsymbol{\Phi}_j\right)_{i,l}. \tag{3.21}$$

- *Number of generated target points is higher than sensor vehicles in neighborhood:* It is not necessary that the number of target points corresponds to the total number of sensor vehicles anymore. This would present a significant and avoidable computation overhead. However, the number of target points should be larger than the maximum number of sensor vehicles within a neighborhood. A good balance enables an adequate monitoring of the local area, coevally pulling the vehicles to regions with higher uncertainty where either additional measurements are required or an exchange with a sensor vehicle residing in this area is possible.

- *Local appealing points remain local entities:* For the local appealing points, no further special treatment is necessary if the error variance $\mathbf{v}_j$ is available. Every sensor vehicle computes the appealing point of itself as well as the local appealing points for all neighbors so that they can be considered in the optimization process.

3.6 Decentralized Cooperative Vehicle Controller

The decentralized monitoring strategy is finalized by the use of a decentralized cooperative vehicle controller. A review of related decentralized cooperative strategies is provided in [127]. However, it is straightforward to apply the extended version of the optimization-based cooperative feedback controller used in Section 2.6 for a decentralized scenario, which has been presented in [104].

In the decentralized variant, each sensor vehicle maintains its own personal cooperative vehicle controller and solves the MLD-based optimal control problem presented in Section 2.6 in a model-predictive control fashion. In contrast to the problem considered in the previous chapter, only the vehicles in the neighborhood N_j are considered for the problem of sensor vehicle j. Thus, the local vehicle state vector $\mathbf{p}_{\mathbf{g},j}^{(k)}$ only consists of the positions, velocities, etc. of vehicle j and all vehicles residing within its communication range d_j, i.e. $\mathbf{p}_{\mathbf{g},j}^{(k)} = \text{col}\{\mathbf{p}_i^{(k)} : i \in N_j\}$. The same applies for the other entities, for example for the output control set $U_{n_{Tr},j}^{(k)}$, which comprises only the control inputs for the vehicles in the neighborhood of vehicle j. Reformulating the centralized MILP (2.83) yields the decentralized MILP

$$\min_{U_{n_{Tr},j}^{(k)}} \; |\mathbf{F}_j \tilde{\mathbf{p}}_j^{(k+n_{Tr})}| + \sum_{i=0}^{n_{Tr}-1} |\mathbf{G_{1}}_j \tilde{\mathbf{u}}_j^{(k+1)}| + |\mathbf{G_{2}}_j \boldsymbol{\delta}_j^{(k+i)}| + |\mathbf{G_{3}}_j \mathbf{z}_j^{(k+i)}| + |\mathbf{G_{4}}_j \tilde{\mathbf{p}}_j^{(k+i)}| \tag{3.22a}$$

$$\text{s.t.} \quad \tilde{\mathbf{p}}_j^{(k+i+1)} = \mathbf{A}_j \tilde{\mathbf{p}}_j^{(k+i)} + \mathbf{B_{1}}_j \tilde{\mathbf{u}}_j^{(k+i)} + \mathbf{G_{2}}_j \boldsymbol{\delta}_j^{(k+i)} + \mathbf{G_{3}}_j \mathbf{z}_j^{(k+i)} \tag{3.22b}$$

$$\mathbf{E_{2}}_j \boldsymbol{\delta}_j^{(k+i)} + \mathbf{E_{3}}_j \mathbf{z}_j^{(k+i)} \leq \mathbf{E_{1}}_j \tilde{\mathbf{u}}_j^{(k+i)} + \mathbf{E_{4}}_j \tilde{\mathbf{p}}_j^{(k+i)} + \mathbf{E_{5}}_j \; . \tag{3.22c}$$

Each sensor vehicle applies only the first element of the resulting control input set $U_{n_{Tr},j}^{(k)}$ it has determined for itself. Potential control inputs calculated for neighboring sensor nodes are not communicated and serve only for assessing the behavior of the neighboring vehicles, whose movement is also based on the minimization of a similar optimal control problem. Obviously, the decentralized strategy cannot compete with the optimality of the trajectories in the centralized approach, but the feedback properties of MPC avoid larger deviations.

In general, the effort for the solution of (3.22) is much lower than the effort required for computing (2.83) since the number of vehicles involved in the computation is much smaller. The decentralized vehicle controller, thus, suits the limited computational capabilities in a decentralized monitoring approach and provides scalability with respect to a higher number of sensor vehicles.

3.7 Implementation, Evaluation and Results

The proposed decentralized monitoring strategy is tested with several parameter settings to evaluate its performance regarding accuracy and scalability. After a brief notion on the implementation of the approach in Subsection 3.7.1, general results and a comparison of the results to the centralized monitoring approach of Chapter 2 are considered in Subsection 3.7.2. Subsection 3.7.3 and Subsection 3.7.4 focus

on the influence of the ROM and the decentralized data fusion, before computation time and scalability are investigated in Subsection 3.7.5.

3.7.1 Implementation

In order to implement the ideas described in this chapter, the MATLAB code of the centralized DDDAS, briefly described in Subsection 2.7.1, has been extended. Several full simulations are conducted to generate the snapshot set, calculate the basis and reduce the model dimension prior to the dynamic data-driven simulation. Furthermore, the decentralized data fusion procedure is implemented using the MATLAB function `fmincon` to solve the optimization problem associated with CI. Concerning the decentralized vehicle controller, it should be obeyed that several possible controller structures are possible since the number of sensor vehicles to be considered is dynamically changing. The possible structures should be set up prior to the simulation so that the necessary structure can be chosen dynamically.

3.7.2 General Results of the Proposed Decentralized Approach

A test case similar to the general test case in Subsection 2.7.2 is set up to show the performance of the proposed decentralized monitoring approach. While the problem domain $\Omega = [0\,\mathrm{m}, 1000\,\mathrm{m}] \times [0\,\mathrm{m}, 1000\,\mathrm{m}]$, stationary process velocity field $w_1(r_1, r_2) = 3 \cdot 10^{-6}\,\mathrm{m^{-1}s^{-1}} \cdot r_2^2$ and $w_2(r_1, r_2) = 2 \cdot 10^{-6}\,\mathrm{m^{-1}s^{-1}} \cdot r_1^2 + 0.5\,\mathrm{ms^{-1}}$ and the constant diffusion coefficient $D = 5\,\mathrm{m^2s^{-1}}$ are adopted, a different initial condition and a source term are considered in this scenario. In particular, the true initial condition (2.85) with

$$\boldsymbol{\Sigma_1} = \begin{pmatrix} 10^{-4} & 0 \\ 0 & 10 \cdot 10^{-4} \end{pmatrix} [\mathrm{m^{-2}}], \quad \boldsymbol{\Sigma_2} = \begin{pmatrix} 10^{-4} & 0 \\ 0 & 10^{-4} \end{pmatrix} [\mathrm{m^{-2}}] \tag{3.23}$$

and $\mathbf{r_1} = (300\,\mathrm{m}, 250\,\mathrm{m})^T$, $\mathbf{r_2} = (400\,\mathrm{m}, 250\,\mathrm{m})^T$ is applied. The true source is governed by (2.90) with $\mathbf{r_s} = (250\,\mathrm{m}, 450\,\mathrm{m})^T$. Furthermore, the true background velocity and diffusivity are perturbed in every step according to the explanations in Subsection 2.7.2.

For the estimation, three potential sources, which may contribute to the true source output, are considered with respective centers at $\mathbf{r}_1 = (250\,\mathrm{m}, 450\,\mathrm{m})^T$, $\mathbf{r}_2 = (350\,\mathrm{m}, 350\,\mathrm{m})^T$ and $\mathbf{r}_3 = (450\,\mathrm{m}, 250\,\mathrm{m})^T$. This means that three source output parameters θ_i are to be estimated besides the state vector. The initial estimate for both, parameters and state vector, are set to zero.

The corresponding error covariance matrices have the form of (2.87) and (2.88) and are listed, besides other settings in Table 3.7.2.

Again, three sensor vehicles are applied starting from the positions $\mathbf{r}_1^{\mathbf{p}} = (300\,\mathrm{m}, 300\,\mathrm{m})^T$, $\mathbf{r}_2^{\mathbf{p}} = (500\,\mathrm{m}, 700\,\mathrm{m})^T$ and $\mathbf{r}_3^{\mathbf{p}} = (700\,\mathrm{m}, 300\,\mathrm{m})^T$. Unlike the application of the previous chapter, the limited communication range of the vehicles is considered in this section. Communication is only possible with vehicles residing within $d = 200\,\mathrm{m}$.

The proposed decentralized dynamic data-driven monitoring requires the generation of a reduced order model. Several offline computations are performed with the original model of $n = 1886$ nodes to build up the snapshot set.

At first, simulations of the impulse response corresponding to the three considered source locations are conducted. Furthermore, a set of possible initial conditions, among them the true initial condition, is considered and simulation results obtained from the runs with the different initial conditions are added to the

Table 3.1: Simulation and controller settings

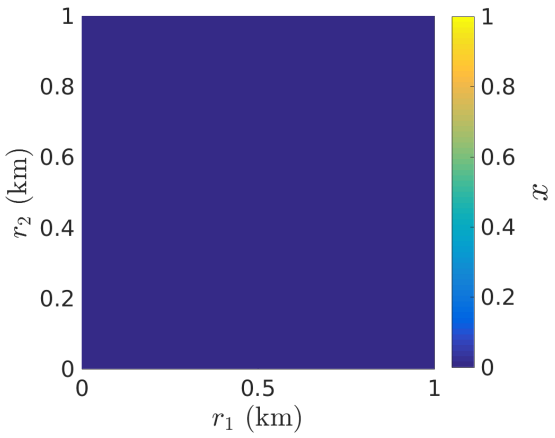
Number of time steps n_T	200
Time step size Δt	2
Observation variance magnitude R_A	0.02
Minimum observation variance R_{min}	$1 \cdot 10^{-3}$
Initial covariance magnitude P_A	0.2
Initial covariance center $\mathbf{r_P}$ [m]	$(250, 250)^T$
Initial covariance area L_{P1} [m^2]	$1.3 \cdot 10^5$
Initial covariance area L_{P2} [m^2]	$2 \cdot 10^4$
Model covariance magnitude Q_A	0.02
Model covariance area L_Q [m^2]	$2 \cdot 10^4$
Initial parameter covariance $P_\theta^{(0)}$	0.01
Parameter model error variance Q_θ	0.0125
Maximum vehicle speed v_{max}^p [ms^{-1}]	10
Maximum vehicle acceleration a_{max}^p [ms^{-2}]	10
MPC horizon n_{Tr}	20
Number of hyperplanes n_γ	8

snapshot set. Although external perturbations in this scenario are chosen to be low, to account for model error, the snapshot is complemented with the impulse response of the first ten dominant eigenvectors of a representative model error covariance matrix $\mathbf{Q}$ (correlation area of $2 \cdot 10^4$ m^2, highest magnitude at $(0.5, 0.5)^T$). Prescribing a nominal value of $\delta = 99.9999\%$ of the snapshot energy to be captured, a ROM with $n_\ell = 91$ basis vectors is obtained.

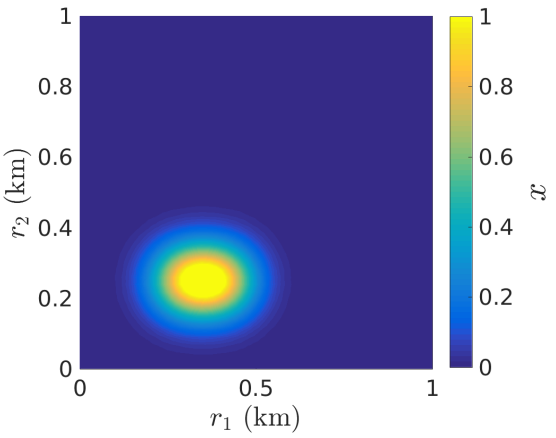
Applying the obtained ROM within the proposed decentralized monitoring approach with the specified settings yields the estimated concentration fields depicted in Figure 3.3. Starting with a bad estimate, the estimate is improved over time and a qualitative difference between the true and the estimated state can hardly be noticed at final time.

The above results (and all following results stemming from the decentralized monitoring approach, if not stated otherwise) are obtained from fusing the individual decentralized estimates according to CI. Due to the limited communication range, exchange of information is sometimes not possible and, generally, different estimates are available at different nodes. Figure 3.4 clarifies this issue. The estimate at $t = 22$ s of every sensor vehicle is illustrated together with the current position of the sensors and the communication range of the respective sensor vehicle. The first sensor vehicle has operated in an informative area and, therefore, maintains a rather good estimate. However, until the considered time stage, no communication has been possible to the other two sensor vehicles, which are currently heading towards the informative area. For this reason, the estimate of the other two sensor vehicles has not been updated yet with the valuable information of Node 1 so that the estimate of both sensor nodes is worse. As the latter two vehicles are located within each other's communication range, they share a common estimate. In the following step, Vehicle 1 will enter the communication range of the two other sensor vehicles so that information is exchanged and estimates can be improved.

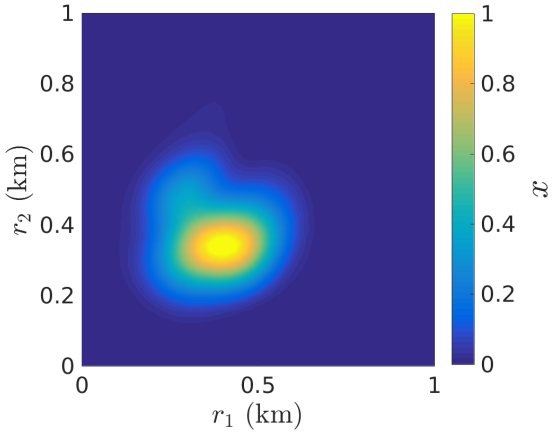
For a quantitative assessment of the proposed decentralized approach, the root mean square error between the estimated process state and the true state is calculated. It is plotted over time in Figure 3.5. A large



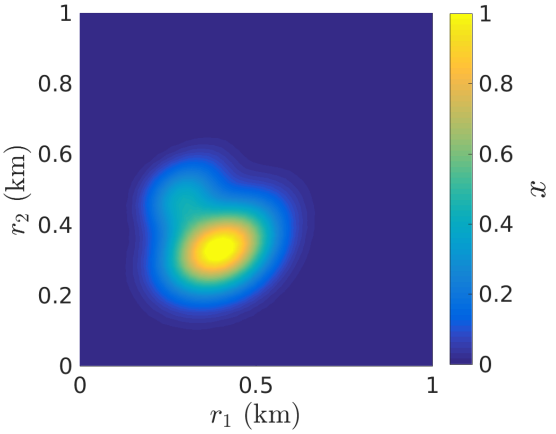
(a) Estimated concentration at $t = 0s$



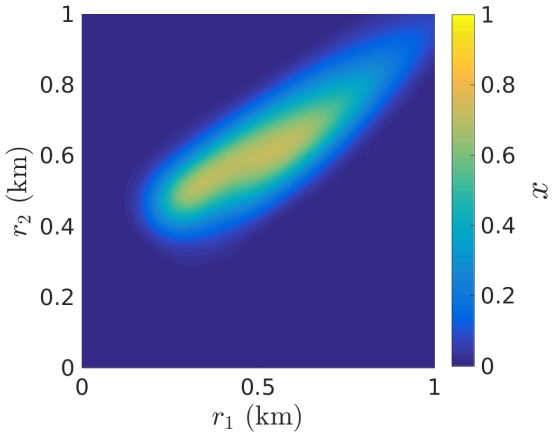
(b) True concentration at $t = 0s$



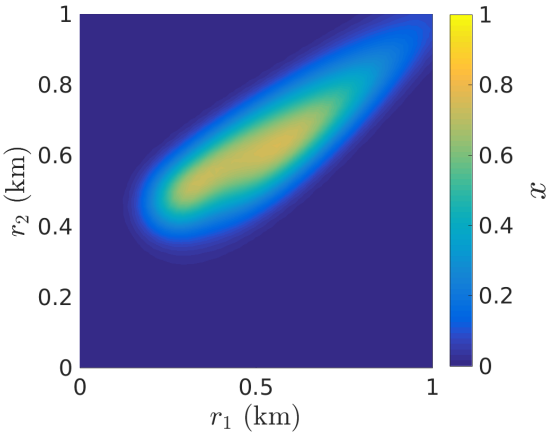
(c) Estimated concentration at $t = 100s$



(d) True concentration at $t = 100s$

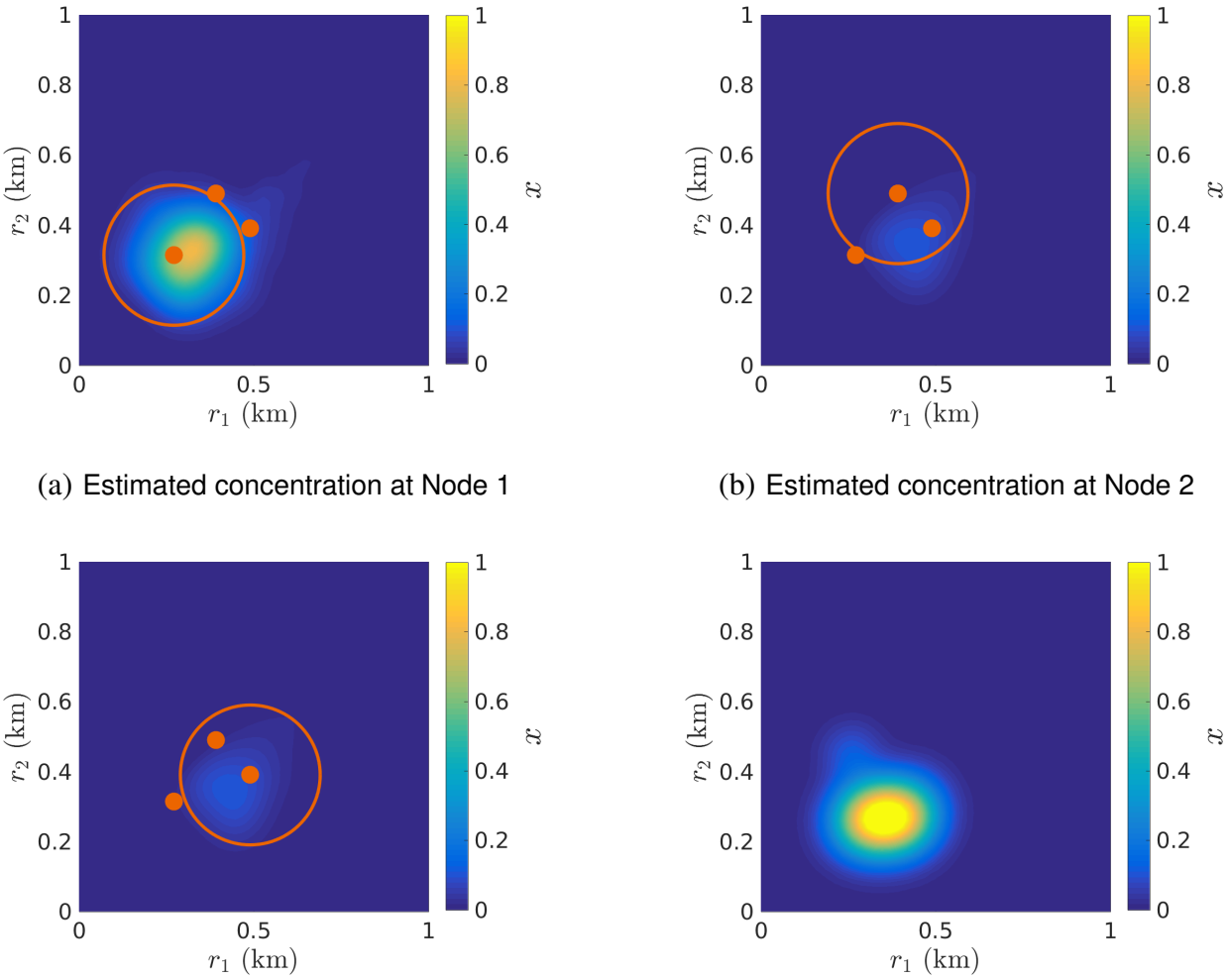


(e) Estimated concentration at $t = 400s$



(f) True concentration at $t = 400s$

Figure 3.3: Concentration distribution at different times – estimate (left) vs. true state (right)



(a) Estimated concentration at Node 1

(b) Estimated concentration at Node 2

(c) Estimated concentration at Node 3

(d) True concentration

Figure 3.4: State estimate at $t = 22\,\mathrm{s}$ at different sensor vehicles vs. true state. The positions of the sensor vehicles are indicated with orange dots, the circles show the communication ranges of the considered vehicles.

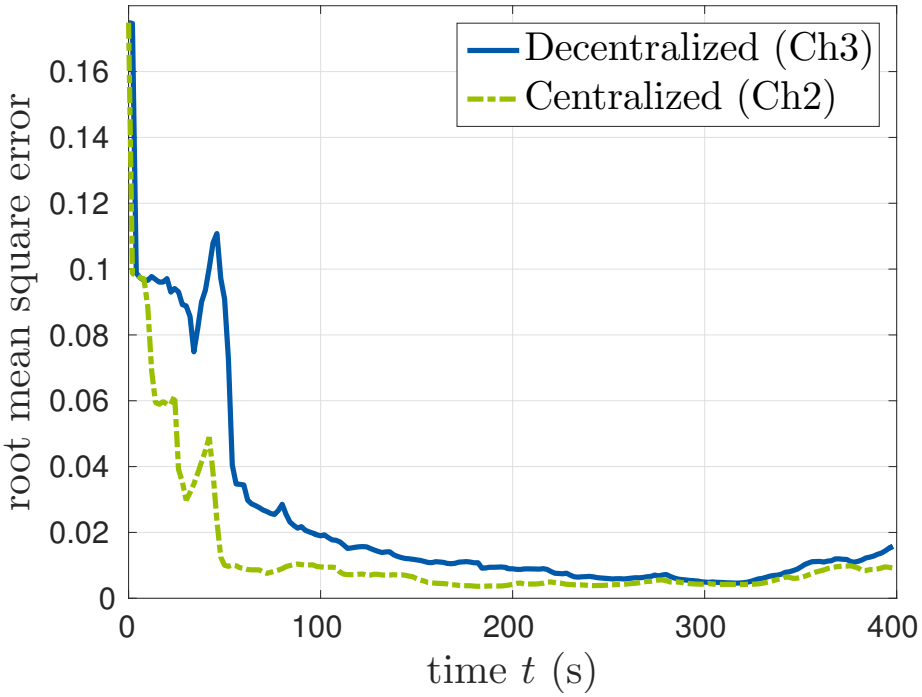


Figure 3.5: Root mean square error of the state estimate obtained with the proposed centralized monitoring approach in Chapter 2 compared to the error of the proposed decentralized monitoring approach of Chapter 3 over time

reduction in error can be noticed, followed by a time phase, in which the error is reduced slightly or kept relatively constant. For comparison, the test case is also performed with the proposed centralized dynamic data-driven monitoring strategy described in Chapter 2. The error reduction coming along with the centralized approach is, of course, larger, especially at early time stages. However, for larger times, the differences in error between decentralized and centralized approach are minor.

The same conclusions can be drawn for the parameter estimate, which is plotted over time in Figure 3.6. While the centralized approach provides a better estimate at earlier time stages, both variants provide a rather good parameter estimate after approximately $200\,\mathrm{s}$.

3.7.3 Investigation of the Influence of Reduced Order Model

The application of a reduced order model drastically reduces the problem dimension and enables the computation of high-dimensional problems on machines with limited computational power. Some properties concerning its generation and application are examined in the following.

POD-based model order reduction is motivated by the fact that it is possible to approximate a set of exemplary process states by a low number of basis vectors. For the test case scenario described in the previous subsection, this issue is illustrated in Figure 3.7, where the normalized eigenvalues corresponding to the first 100 eigenvectors are plotted. Eigenvectors with a large eigenvalue capture a high energy of the snapshot set and contribute much to its approximation, whereas eigenvectors with small eigenvalues only have a small influence on the representation. Evidently, the eigenvalues decay exponentially so that a small number n_ℓ of basis vectors is already sufficient for a good quality representation of the snapshot set. Furthermore, similar decays can be noticed for models with different original state dimension n. Thus, n_ℓ is almost independent of the number of original grid points enabling high scalability.

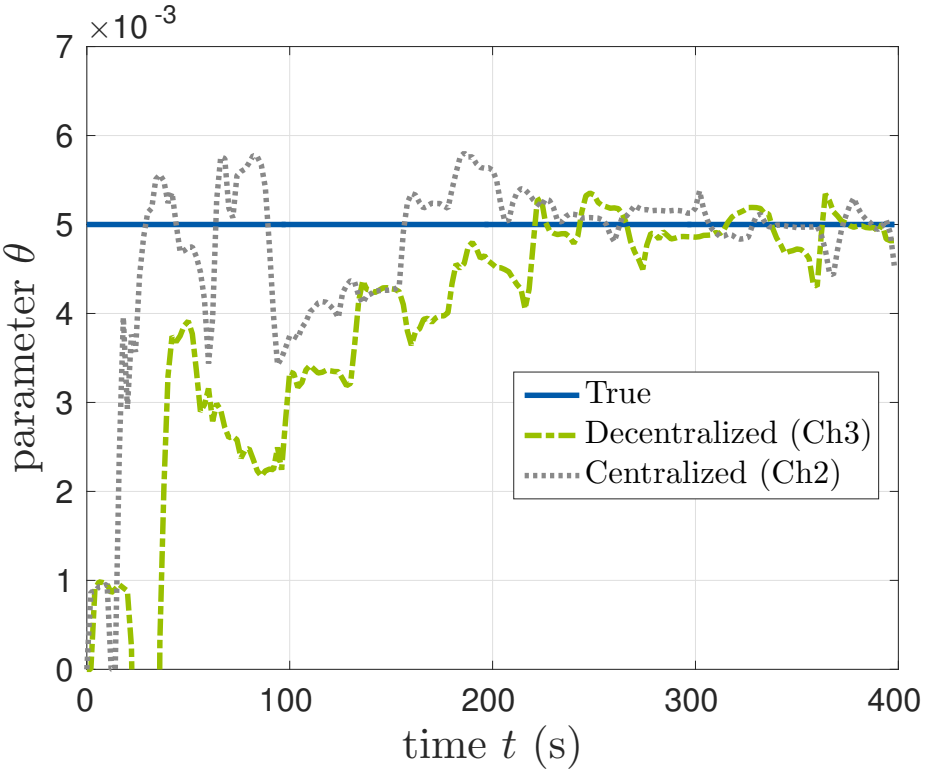


Figure 3.6: Estimate of the source output function θ over time – centralized monitoring approach (cf. Chapter 2) vs. decentralized monitoring approach (cf. Chapter 3)

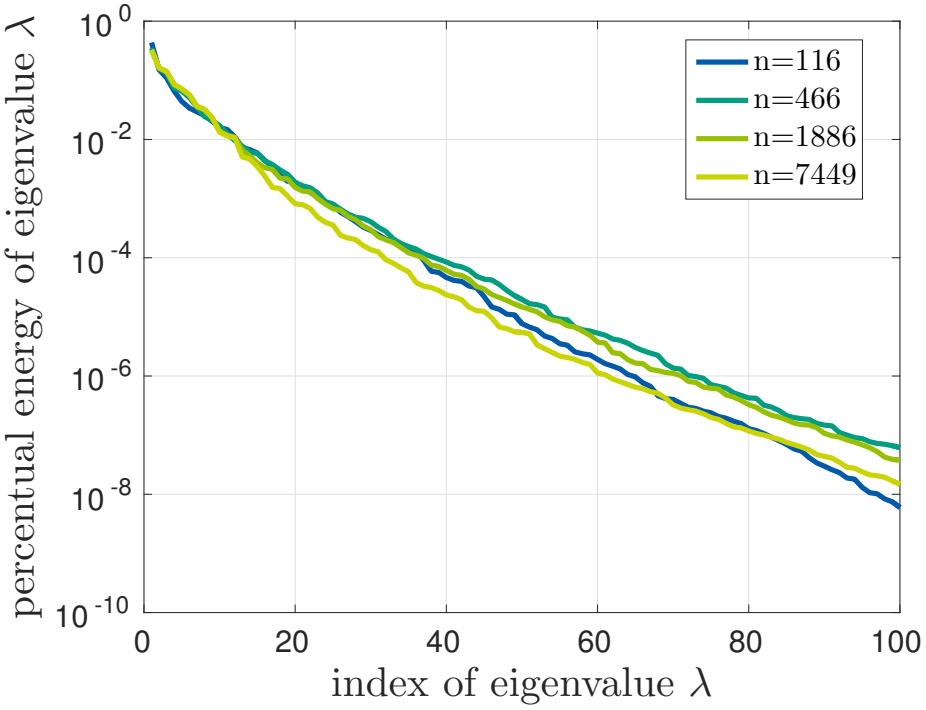


Figure 3.7: Fraction of the captured energy from the total snapshot energy for the first 100 eigenvectors and for different numbers of grid points n

According to (3.5), the number n_ℓ of POD basis vectors can be found by defining a value δ such that the chosen basis captures the fraction δ of the total snapshot energy. For the considered problem, the corresponding results for different δ and different n are listed in Table 3.2. As stated before, n_ℓ is nearly independent of the original problem dimension n and is, even for high δ, relatively low.

Table 3.2 also shows the influence of the number n_η, which describes the number of impulse response simulations for the eigenvectors of a representative model error covariance matrix added to the snapshot set (with a setting of $\mathbf{Q}$ as in the previous subsection). Obviously, more information added to the snapshot set lead to a reduced model with a larger dimension.

Table 3.2: Number n_ℓ of reduced order basis elements in dependence of covered energy fraction δ, original problem dimension n and number n_η of eigenvectors of $\mathbf{Q}$ considered for snapshot set

	$\delta = 0.99$			$\delta = 0.9999$			$\delta = 0.999999$		
	$n_\eta = 0$	$n_\eta = 10$	$n_\eta = 20$	$n_\eta = 0$	$n_\eta = 10$	$n_\eta = 20$	$n_\eta = 0$	$n_\eta = 10$	$n_\eta = 20$
$n = 466$	13	21	32	26	53	80	43	100	150
$n = 1886$	15	20	27	29	51	78	41	91	145
$n = 7449$	15	16	20	29	43	66	43	82	133

Table 3.3: Average relative projection error for a test case with low perturbations (in %)

	$\delta = 0.99$			$\delta = 0.9999$			$\delta = 0.999999$		
	$n_\eta = 0$	$n_\eta = 10$	$n_\eta = 20$	$n_\eta = 0$	$n_\eta = 10$	$n_\eta = 20$	$n_\eta = 0$	$n_\eta = 10$	$n_\eta = 20$
$n = 466$	5.6	4.8	4.2	1.2	0.8	0.6	0.5	0.4	0.3
$n = 1886$	5.5	4.9	4.1	1.4	0.8	0.5	0.9	0.3	0.2
$n = 7449$	5.7	5.5	4.6	1.6	0.9	0.6	0.9	0.4	0.2

Table 3.4: Average relative projection error for a test case with strong perturbations (in %)

	$\delta = 0.99$			$\delta = 0.9999$			$\delta = 0.999999$		
	$n_\eta = 0$	$n_\eta = 10$	$n_\eta = 20$	$n_\eta = 0$	$n_\eta = 10$	$n_\eta = 20$	$n_\eta = 0$	$n_\eta = 10$	$n_\eta = 20$
$n = 466$	17.0	9.2	7.5	12.7	2.9	1.1	8.6	1.2	0.6
$n = 1886$	31.6	18.2	16.8	27.1	4.1	2.1	24.9	1.7	0.5
$n = 7449$	45.1	43.7	28.4	40.9	9.2	5.2	38.7	3.1	1.0

The resulting ROM is tuned for simulations similar to the data provided by the snapshot set. However, practical problems are governed by perturbations and uncertain parameters leading to states that cannot be represented by the snapshot set anymore.

To investigate the quality of the ROM proposed above, a full simulation representing an exemplary true state is conducted. As in the previous test cases, external perturbations are represented by perturbations in process diffusivity and velocity. At first, a scenario with low perturbation (velocity is perturbed by $1\,\mathrm{ms}^{-1}$ per step, filtered with moving average filter; diffusivity is perturbed by $0.01\,\mathrm{m}^2\mathrm{s}^{-1}$ per step) is considered.

The time-averaged relative error

$$e = \frac{1}{n_T + 1} \sum_{k=0}^{n_T} \frac{\left\| \mathbf{\Phi}\mathbf{\Phi}^T \mathbf{x}^{\mathbf{t}(k)} - \mathbf{x}^{\mathbf{t}(k)} \right\|}{\left\| \mathbf{x}^{\mathbf{t}(k)} \right\|} \tag{3.24}$$

between the true solution and the the true solution projected onto the reduced order model provides information about the quality of the reduced model in relation to the considered application.

Table 3.3 contains the respective relative projection error for each considered setting of ROM. Generally, it can be stated that the resulting error is very low and a rather good result can be obtained by the use of almost every presented ROM setting. Increasing the energy fraction δ, naturally, leads to better approximations. The same applies for an increase of n_η: The higher n_η, the less the approximation error e. However, it has to be mentioned that the decrease of error with increasing n_η is only marginally, while the ROM dimension n_ℓ increases significantly. Indeed, considering scenarios with almost identical n_ℓ, e.g. $n = 1886$ with $n_\eta = 20$, $\delta = 0.99$ and $n_\eta = 0$, $\delta = 0.9999$, the use of impulse response snapshots for model error influences is questionable. For almost equal n_ℓ, better results are obtained when perturbations are not considered in the snapshot set.

However, the situation is different if stronger perturbations are considered. An exemplary test run with a strong perturbation (velocity is perturbed by $2\,\mathrm{ms}^{-1}$ per step, filtered with moving average filter; diffusivity is perturbed by $0.5\,\mathrm{m}^2\mathrm{s}^{-1}$ per step) leads to the results of Table 3.4. The relative errors are much higher and the ROM obtained from the standard snapshot set consisting of source impulse response and initial condition snapshots is not capable of accurately approximating the true process state, even if δ is increased. However, the application of the proposed model error snapshots provides a much better approximation of the true state, although the actual behavior of the model error has not be known a priori.

All things considered, the snapshot set build up by a combination of initial condition and source impulse response snapshots provides a suitable choice when the model error, e.g. the external perturbations are low. Considering stronger perturbations, the snapshot set should be complemented with the proposed model error impulse response snapshots. Even if the model error is not Gaussian and known exactly, much better results can be obtained.

The results in the following are all mainly focused on low perturbations. If the ROM is applied for simulation of the perturbed problem, the results depicted in Figure 3.8 are obtained. Due to accumulating model error, which cannot be properly represented by ROM, the error increases on average over time. For the considered applications, δ should be chosen to be larger than 99% to obtain an accurate representation. In this case, the resulting error stemming from model reduction only contributes slightly to the total estimation error (cf. Figure 3.5, where a much lower perturbation has been used).

3.7.4 Investigation of the Influence of Limited Communication Ranges

Besides the use of a reduced order model, limited communication ranges requiring decentralized data fusion contributes to a larger error when using the decentralized monitoring strategy instead of a centralized one (see Figure 3.5).

Therefore, the influence of different communication ranges on the results obtained with the dynamic data-driven monitoring approach is investigated. Using the test case setting described in the previous subsections with different communication ranges yields the root mean square error depicted in Figure 3.9.

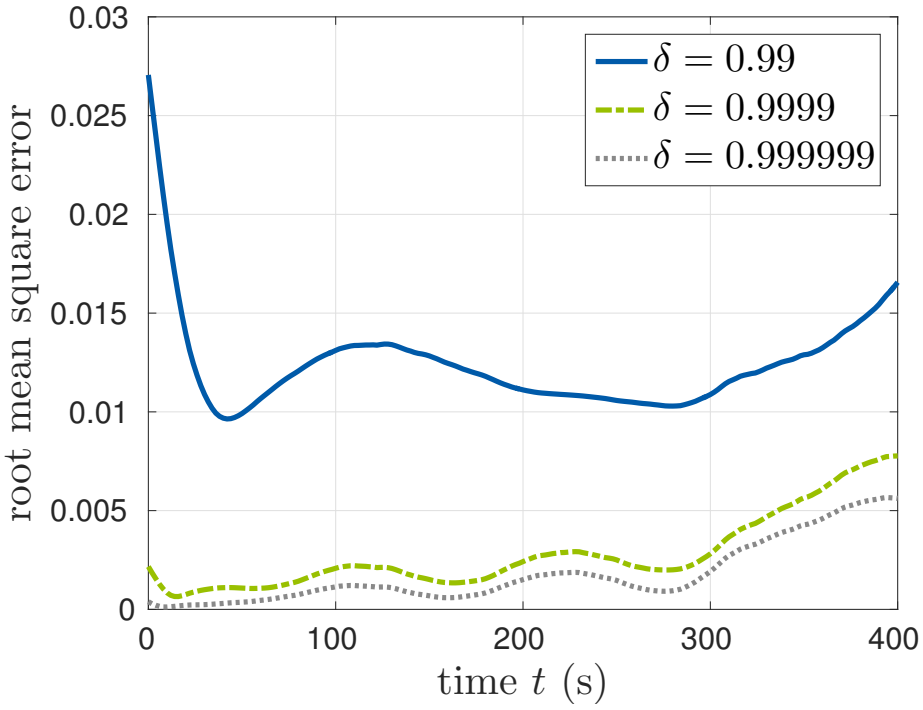


Figure 3.8: Root mean square error of reduced simulation results for different energy fractions δ over time

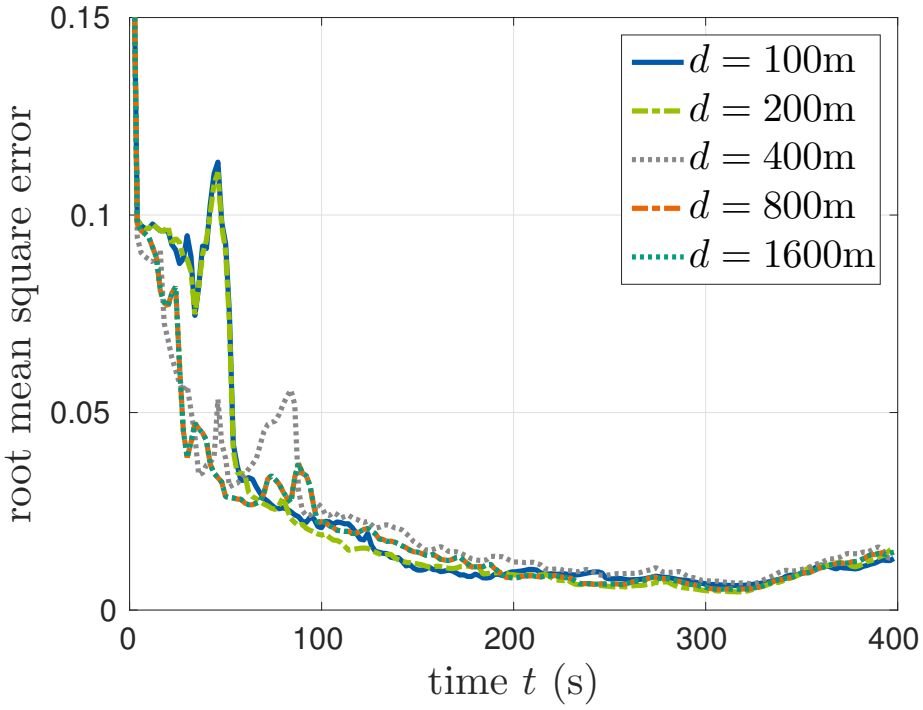


Figure 3.9: Root mean square error of the state estimate obtained with the proposed decentralized approach with different communication radii d over time

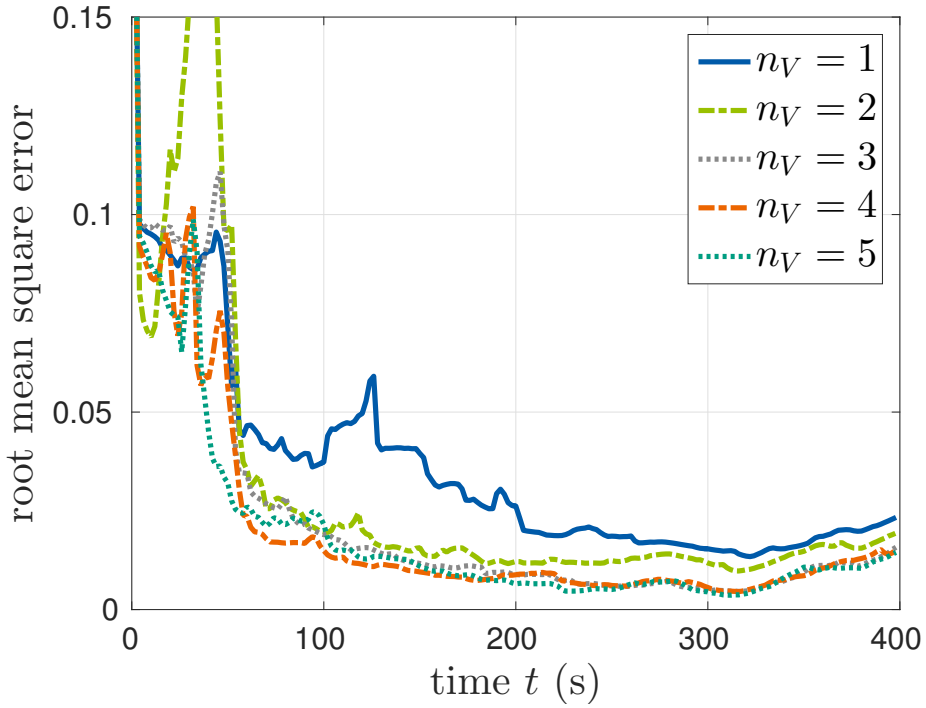


Figure 3.10: Root mean square error of the state estimate obtained with the proposed decentralized approach with a different number of sensor vehicles n_V over time

Larger communication ranges naturally perform better than small communication ranges. However, after approximately $100\,\mathrm{s}$, the obtained results are nearly equivalent. A stronger perturbation of the model state might potentially lead to a more dominant performance of the large range approaches. Using $d = 1600\,\mathrm{m}$, the vehicles are always located in each other's communication range and information exchange between all vehicles can be guaranteed in each step. However, the solution obtained with this approach still does not correspond to the solution of a centralized approach since CI is used for data fusion, which is usually outperformed by Kalman Filter methods.

Moreover, the proposed decentralized approach is tested with a different number of sensor vehicles. A larger number of sensor vehicles does not only lead to a higher number of measurements but also means a better coverage of the considered problem area and an increased information exchange between sensor vehicles.

Figure 3.10 shows the error of the state estimate over time for a different number of sensor vehicles. As already recognized when examining the centralized approach (see Figure 2.18), more sensors lead to an improved estimate and an additional sensor vehicle gets less valuable with an increasing number of total sensor vehicles.

3.7.5 Computation Time and Scalability

Decentralized algorithms meant to be performed on-board the sensor vehicles need to be efficient and are prohibited to be time-consuming. Therefore, the computation time of the test case scenario presented in Subsection 3.7.2 is evaluated with different choices of the number of grid points n, the energy fraction covered by the POD basis δ, and the number of applied sensor vehicles. Once more, it has to be stated that computation time can be further reduced by optimizing the code, but the results (again obtained with

an Intel (R) Core (TM) i7-3820 3.60 GHz, 16 GB RAM) admit an assessment of scalability and potential applicability of the monitoring approach.

Figure 3.11 shows the computation time for one forecast and assimilation step for different numbers of grid points of the original problem n with varying energy fractions δ, which are related to a different dimension of the ROM. Due to the use of reduced order models, calculations are performed on problems with a drastically reduced dimension. This results in only a slight increase of computing time with increasing n. Hence, the proposed approach can be stated to be scalable with respect to larger problem domains and appears to be applicable on machines with lower computational power. Compared to the results of the centralized approach in Figure 2.25 and 2.26, a large speedup is obtained, especially regarding high-dimensional problems.

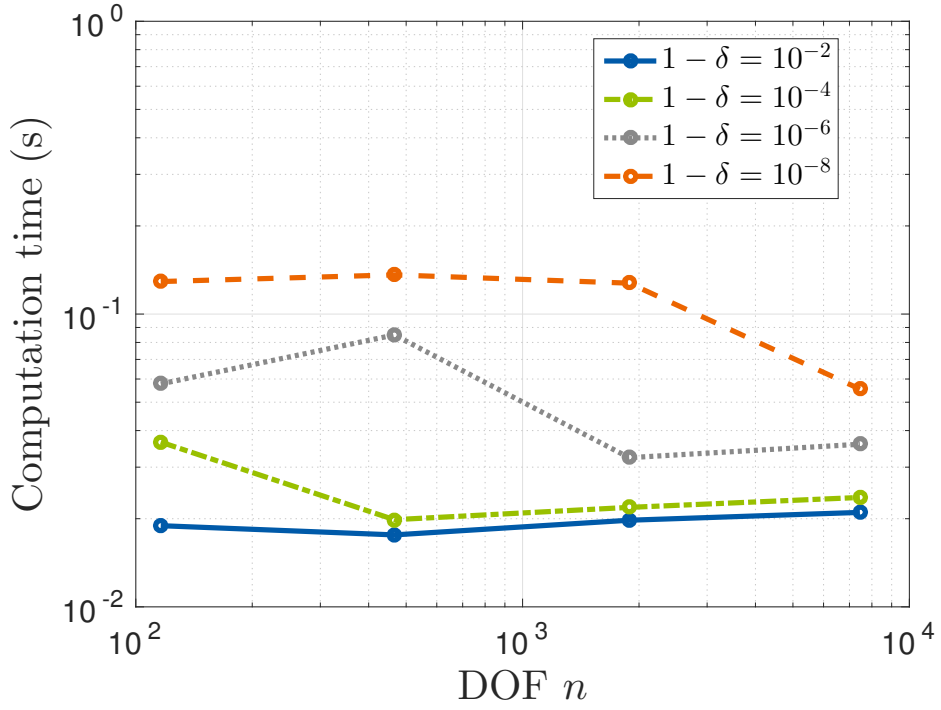


Figure 3.11: Average computation time for one forecast and assimilation step of the proposed decentralized monitoring approach in dependence of original problem dimension n and energy fraction δ covered from snapshot set

Furthermore, computation times for different numbers of sensor vehicles are evaluated. The strategy of the decentralized vehicle control approach is to simplify the optimal control problem by only considering vehicles residing within the local communication range. Thus, a smaller communication range comes along with a less time-consuming determination of the optimal control. Figure 3.12 shows exactly this behavior. However, for $d > 200\,\mathrm{m}$, the sensor vehicles appear to be connected to all other vehicles for the most time so that nearly every time all sensor vehicles are involved in the computations leading to a higher computation time. In total, the benefit of this strategy will be most dominant when applying a moderate number of sensor vehicles with small communication range on a large problem domain. For a lot of sensor vehicles on a small domain, the considered approach cannot be regarded to be scalable and would not at all be real-time capable.

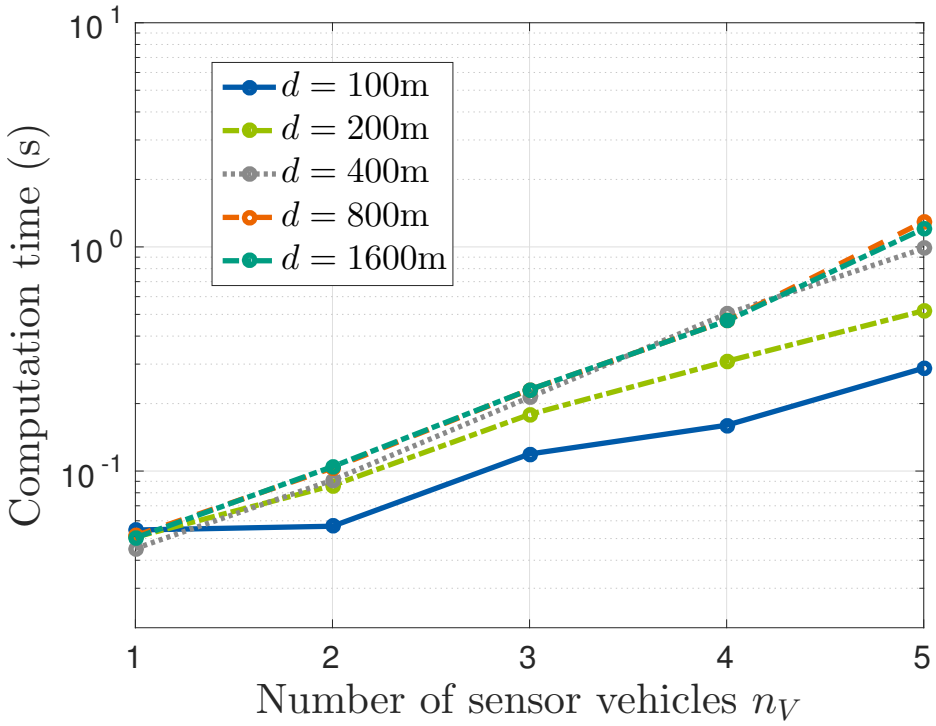


Figure 3.12: Average computation time for one vehicle control step of the proposed decentralized monitoring approach in dependence of number of sensor vehicles n_V and communication radius d

3.8 Summary and Discussion of Chapter 3

A new decentralized dynamic data-driven monitoring approach has been presented in this chapter. Instead of performing most of the required computations on a central supercomputer and broadcasting results to the sensor vehicles, local information is processed onboard the sensor vehicles and exchanged only with sensor vehicles in the vicinity. Compared to the centralized approach presented in the previous chapter, the decentralized approach explicitly takes into account limitations concerning communication ranges and provides more scalability regarding larger problem domains and a higher number of sensor vehicles.

Despite the limited onboard computation capabilities, one does not have to waive PDE-based forecast models and their physically realistic forecast ability. Proper Orthogonal Decomposition is used to generate a variant of the forecast model with reduced dimension, potentially also applicable on computing devices with limited computational power. A suitable choice of the snapshot simulations to build up the reduced order model has been proposed and evaluated.

Based on the reduced order model, a reduced data assimilation procedure fitting the requirements of a decentralized computing structure has been developed. The data assimilation procedure consists of two steps, being based on Kalman Filter and Covariance Intersection. A decentralized extension of the optimization-based feedback controller is employed for vehicle control and fed with target points obtained from a modified version of the sequential target point generator proposed in Chapter 2.

The decentralized monitoring approach has shown a good performance in the provided test cases with estimation results only slightly worse than the results obtained with a centralized version. More specifi-

cally, for the considered test cases with a low perturbation, the estimation error of both, centralized and decentralized approach, have shown to be almost identical after a certain amount of simulation steps. However, it takes a longer time using the decentralized approach to reach a certain threshold in the beginning. In return, it has been shown that the scalability of the decentralized approach is much better providing reduced computing times for the vehicle controller and drastically reduced computing times for the model forecast.

The computers on-board the sensor-carrying vehicles maintain models comprising the whole problem domain, although they are only specialized in the part of the domain they are currently working on. Therefore, a partition of the global domain, as described in the next chapter, appears to be suitable, especially for problem domains with larger spatial dimensions.

Another question that has not been treated is the fusion of the local estimates to get a global estimate at specific times and places. Occasional communication with a base station or the collection of information by a data mule could be possibilities.

4 A New Partitioned Approach to Decentralized PDE-based Dynamic Data-Driven Monitoring

A further novel decentralized dynamic data-driven monitoring approach for atmospheric dispersion processes based on the forecasts of a PDE model is presented in this chapter. Compared to the approach presented in the previous chapter, the global domain is partitioned into several subdomains, on which a group of sensor vehicles is acting. Communicating with sensor vehicles on neighboring domains, the global state vector or error covariance matrix does not have to be maintained, but only the lower dimensional local variant of them. Furthermore, only sensor vehicles on the own domain have to be involved in the vehicle control step. Hence, the partitioned approach further pursues the goal of scalability with respect to larger problem domains and a larger number of sensor vehicles.

The methods that constitute the monitoring approach have to provide results that approximate the results on the full domain, although working on the partitioned domain. For this reason, an efficient domain decomposition method is used for process prediction, a new partitioned ensemble method is proposed for data fusion and a modified vehicle controller is presented that allows the movement of sensor vehicles from one domain into another.

The basics of the work that is presented in this chapter have also been submitted for publication in [149]. A more detailed presentation of the applied methods as well as a more detailed evaluation is provided here.

The approach proposed in this chapter extends the work presented in Chapter 2 and, to a certain degree, the work presented in Chapter 3. For this reason, the outline of this chapter follows the outline of the previous chapters and the reader is referred to Chapters 2 and 3 for already presented foundations. In Section 4.1, based on a short presentation of related strategies, the contribution of this chapter is stated and an overview of the proposed approach is given. An explanation of the applied and developed methods is subject of Sections 4.2-4.6. Section 4.7 presents several test cases that show the good performance as well as the scalability the partitioned approach offers. A summary of the proposed approach including its benefits and applicability is drawn up in Section 4.8.

4.1 A New Partitioned Monitoring Strategy: Contribution and Overview

The method proposed in this chapter is also aimed to estimate the state and important parameters of an atmospheric dispersion process in a decentralized dynamic data-driven fashion. Compared to the monitoring approach presented in Chapter 3, the approach proposed in this chapter should be even more scalable with respect to problem domains with a larger spatial extent. While the sensor vehicles of the approach in the previous chapter maintained models comprising the complete problem domain, now, the domain is decomposed into several subdomains with sensor vehicles operating on assigned subdomains maintaining only models of their local subdomain. Moreover, communication is only necessary and possible with vehicles operating on the same or neighboring domains. Thereby, computation time and memory associated with model operations as well as the computational effort required for incorporating the effect of other sensor vehicles (regarding e.g. cooperative vehicle control) can be reduced significantly. Consequently, the monitoring approach becomes more scalable with respect to a larger problem

domain and a higher number of sensor vehicles. Besides, the decentralized approach benefits from lower communication requirements and robustness compared to related centralized approaches.

To the best of the author's knowledge, a decentralized dynamic data-driven monitoring on partitioned domains using mobile sensor vehicles has not been considered yet. Most work available in related fields treats decentralized fusion in partitioned systems. While several approaches have been proposed considering sparsely connected systems [100, 165, 61, 97], literature on decentralized fusion for partitioned models resulting from the discretization of a PDE is relatively sparse. In [129], a fusion strategy for state vectors of unequal size is investigated, whereas measurements at the interface between two subdomains are used to complement a simulation on decomposed domains in [143, 144]. A distributed FEM-based Kalman Filter for use in partitioned systems is only considered in [13]. However, using this approach, the error covariance matrix is not properly propagated from one subdomain to another and overlapping subdomains are considered instead of non-overlapping subdomains required for the approach presented here. Furthermore, this approach is not meant to be dynamic data-driven so that only a static sensor network is considered.

Based on this research gap, the main contribution of this chapter is the presentation of a novel decentralized dynamic data-driven monitoring strategy working on decomposed domains. A PDE-based forecast method provides physically realistic predictions, which can be obtained cost-efficiently due to the partitioning of the global domain. All further computations, data fusion as well as cooperative vehicle control, also benefit of the decomposition regarding efficiency.

Again, a number of different building blocks are necessary to set up the complete monitoring strategy. Extending the work presented in the previous chapters, it has to be ensured that all the methods work with respect to a partitioned problem structure. In particular, the interaction between subdomains has to be investigated to guarantee that an approximation of the solution of multiple local problems leads to the approximate solution of the global problem.

Figure 4.1 provides an overview of the proposed approach. In the remainder of this chapter, every building block is explained regarding modifications and innovations with respect to the new problem structure.

First, the decomposition of the domain and the forecast model are described in Section 4.2. Several non-overlapping subdomains are constructed whose union yields the global domain. The forecast equations are solved locally on every domain. However, passing of information between the subdomains is necessary so that the correct boundary conditions can be set. With the use of damped Adaptive Dirichlet-Neumann (d-ADN), a domain decomposition method from literature, and the assumption of certain properties of the velocity field, the global forecast problem can be solved in an efficient manner.

Compared to Chapter 3, the communication topology as well as the observation model has to be slightly modified to account for the partitioning. Section 4.3 describes the new topology together with its benefits.

The next step consists of the fusion of measurements and forecast. Section 4.4 explains the challenges of data assimilation in the considered context and, thereupon, presents a new ensemble-based estimation strategy for decomposed domains. This strategy implicitly includes localization and avoids an exchange of large data sets. Instead, only low-dimensional vectors and matrices have to be exchanged with sensor vehicles on neighboring domains so that measurements can naturally influence estimates on neighboring domains.

Based on the resulting estimates, important measurement locations have to be identified. The identification strategy of the previous chapters is extended for the partitioned approach to allow the sensor vehicles to move from one domain to another. In this way, the number of sensors per subdomain can be

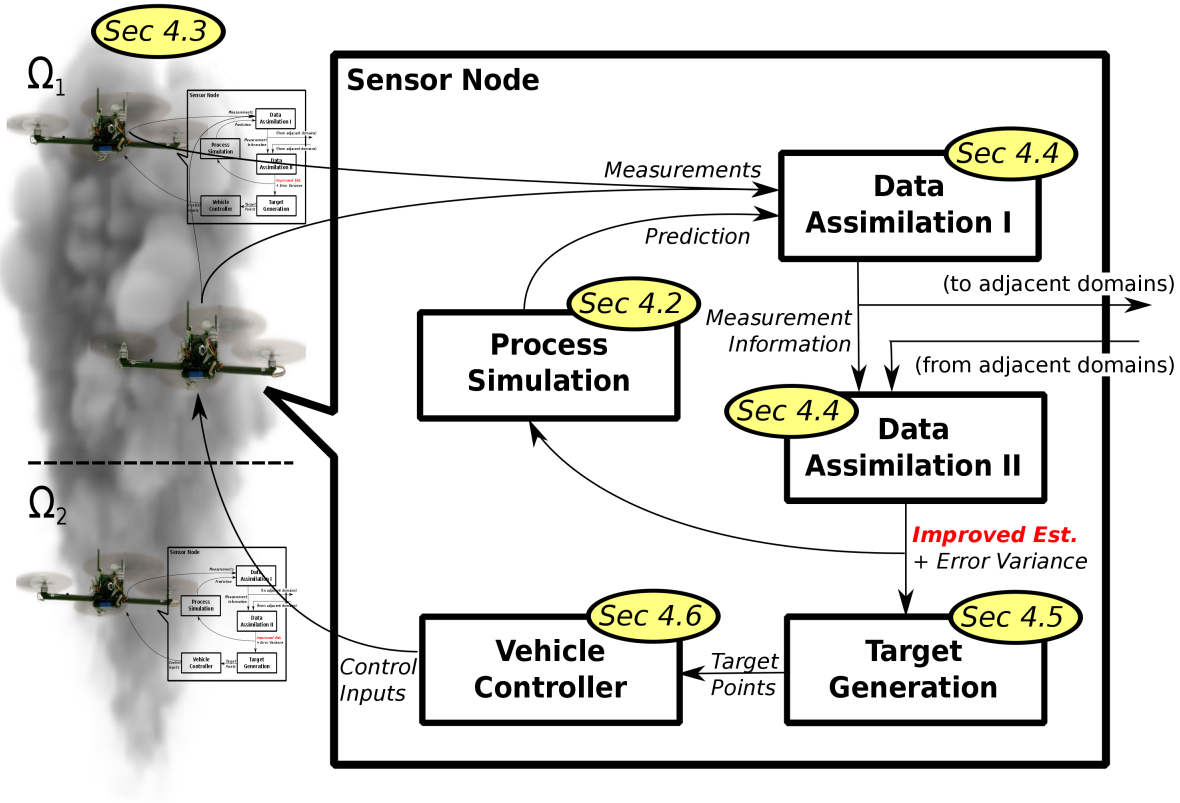


Figure 4.1: Overview of the proposed partitioned monitoring approach

dynamically adapted based on the current cumulative uncertainty, each subdomain is currently coping with. The new target point procedure is subject of Section 4.5.

Finally, the adapted structure of the decentralized, cooperative, optimization-based feedback vehicle controller is briefly described Section 4.6.

4.2 Modeling and Simulation on Partitioned Domains

Motivated by the desire of further scalability, the problem domain is partitioned into several subdomains. This has the advantage that the solution of the forecast equation on the subdomains can be obtained much faster. However, to ensure convergence, exchange between subdomains is necessary.

In the following, it is explained how the domain is decomposed (Subsection 4.2.1) and how an approximation of the global solution is efficiently obtained by the use of d-ADN for solving multiple subproblems (Subsection 4.2.2).

4.2.1 Decomposition of the Global Domain

The global domain Ω is divided into n_d subdomains Ω_i with

$$\Omega = \cup_{i\in\{1,\dots,n_d\}}\Omega_i. \tag{4.1}$$

In the context of this work, a non-overlapping domain decomposition is applied since each group of sensor vehicles should be responsible for a specific, fixed part of the domain. This means that the domains overlap only at their boundary, i.e. $\Gamma_{ij} = \Omega_i \cap \Omega_j = \partial\Omega_i \cap \partial\Omega_j$ denotes the boundary between subdomains Ω_i and Ω_j. An exemplary decomposition is depicted in Figure 4.2.

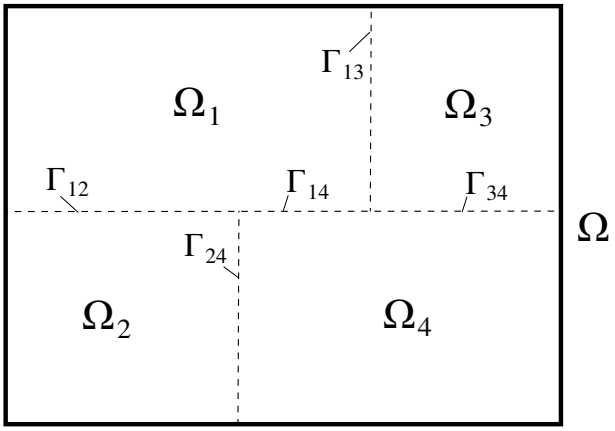


Figure 4.2: Exemplary domain decomposition

Obviously, the problem dimension of the local subdomains is much smaller than the one of the original domain facilitating numerical computations on them.

Every sensor vehicle is assigned to the subdomain Ω_i it is currently located in and is meant to perform computations related to this subdomain. Following this idea, a process model of the local subdomain is maintained at every vehicle on the subdomain.

4.2.2 Process Model and Simulation on Partitioned Domain

To set up the process model of the local subdomain, the original initial boundary value problem (2.7) has to be reformulated. Instead of one large problem for $x(\mathbf{r}, t)$ on Ω, multiple smaller problems for $x_i(\mathbf{r}, t)$ on Ω_i with $i \in \{1, \ldots, n_d\}$ have to be solved

$$\frac{\partial x_i}{\partial t} + \mathbf{w} \cdot \nabla x_i - D\Delta x_i = s \quad \text{in } \Omega_i \tag{4.2a}$$

$$x_i = x_{s,i} \quad \text{on } \partial\Omega_D \tag{4.2b}$$

$$\nabla x_i \cdot \mathbf{n} = t_{s,i} \quad \text{on } \partial\Omega_N \tag{4.2c}$$

$$x_i = x_j \quad \text{on } \Gamma_{ij} \quad j \in \{1, \ldots, n_d\} \tag{4.2d}$$

$$\nabla x_i \cdot \mathbf{n} = \nabla x_j \cdot \mathbf{n} \quad \text{on } \Gamma_{ij} \quad j \in \{1, \ldots, n_d\} \tag{4.2e}$$

$$x_i(\mathbf{r}, 0) = x_i^{(0)}(\mathbf{r}) \quad \text{in } \Omega_i. \tag{4.2f}$$

The transmission conditions (4.2d) and (4.2e) demand a continuous and smooth transition at the boundaries Γ_{ij} and guarantee convergence of the global problem. They provide the coupling between the local solutions on the subdomains and a specific strategy has to be applied to fulfill them.

In general, several different algorithms have been proposed in the field of domain decomposition to solve (4.2). Domain decomposition for speeding up simulations is, in fact, a topic highly investigated

in literature [141, 180]. The strategy is frequently applied for preconditioning a large system or for parallelization.

Here, as described in Section 4.1, the problem is solved using several non-overlapping subdomains Ω_i leading to the partitioned initial boundary value problem (4.2). A popular method to solve such kinds of problems is the Dirichlet-Neumann method [141]. While problem (4.2) is solved on subdomain Ω_i with a Dirichlet condition (4.2d) at boundary Γ_{ij}, problem (4.2) is solved on subdomain Ω_j with a Neumann condition (4.2e) at Γ_{ji}. Several iterations using the output of the solution on the other domain for an updated boundary condition are intended to provide convergence of the global problem.

Neumann conditions should typically be used on outflow boundaries and Dirichlet conditions on inflow boundaries since Dirichlet conditions on outflow boundaries can produce artificial layers in the solution. Therefore, Adaptive Dirichlet-Neumann (ADN) [64] divides the boundaries into parts in which the flow is directed into the subdomain and into parts, in which the flow is directed out of the subdomain

$$\Gamma_{ij}^{\text{in}} = \{\mathbf{r} \in \Gamma_{ij} : \mathbf{w}(\mathbf{r}) \cdot \mathbf{n}(\mathbf{r}) < 0\} \tag{4.3}$$

$$\Gamma_{ij}^{\text{out}} = \{\mathbf{r} \in \Gamma_{ij} : \mathbf{w}(\mathbf{r}) \cdot \mathbf{n}(\mathbf{r}) \geq 0\}, \tag{4.4}$$

where the normal unit vector $\mathbf{n}$ points out of Ω_i. The type of the boundary describes the type of boundary condition that is set.

Two assumptions can be made that drastically reduce the communication and computation effort for solving (4.2). First, the flow over every boundary Γ_{ij} is only directed in one direction so that Γ_{ij} can be completely considered to be either of type Γ_{ij}^{out} or of type Γ_{ij}^{in} for subdomain Ω_i and vice versa for Subdomain Ω_j. Second, as stated before, the problem is considered to be highly advection-dominated. In this case, a further simplification becomes possible as damped Adaptive Dirichlet-Neumann (d-ADN) can be applied [30, 64].

Instead of using the prescribed Neumann condition, a homogeneous Neumann condition is used on Γ_{ij}^{out} so that coupling between the solutions on the different domains is dissolved. As a result, the following problem has to be solved on every subdomain

$$\frac{\partial x_i}{\partial t} + \mathbf{w} \cdot \nabla x_i - D\Delta x_i = s \quad \text{in } \Omega_i \tag{4.5a}$$

$$x_i = x_{s,i} \quad \text{on } \partial\Omega_D \tag{4.5b}$$

$$\nabla x_i \cdot \mathbf{n} = t_{s,i} \quad \text{on } \partial\Omega_N \tag{4.5c}$$

$$x_i = x_j \quad \text{on } \Gamma_{ij}^{\text{in}} \tag{4.5d}$$

$$\nabla x_i \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_{ij}^{\text{out}} \tag{4.5e}$$

$$x_i(\mathbf{r}, 0) = x_i^{(0)}(\mathbf{r}) \quad \text{in } \Omega. \tag{4.5f}$$

This problem can be solved using FEM on every subdomain as described in Subsection 2.2.3. To account for the Dirichlet condition (4.5d), the part $\mathbf{x}_i^{\Gamma_{ij}(k)}$ of the state vector $\mathbf{x}_i^{(k)}$ at time $t = t^{(k)}$ corresponding to the boundary nodes on Γ_{ij}^{out} has to be sent to the vehicles in the neighboring subdomain Ω_j.

As a result of the simulation, the composed state vector

$$\tilde{\mathbf{x}}^{(k)} = \text{col}\{\mathbf{x}_i^{(k)}\}_{i=1}^{n_d} \tag{4.6}$$

is obtained. Compared to the global n-dimensional state vector $\mathbf{x}^{(k)}$ introduced in Chapter 2, $\tilde{\mathbf{x}}^{(k)}$ is potentially permuted and contains the nodal values at the subdomain boundaries twice.

4.3 Modeling of Sensor Vehicles on Partitioned Domains

Like the approach presented in Chapter 3, the sensor network considered in this chapter is assumed to be decentralized. However, the actual communication topology and the observation model differ from the topologies and models that were used before since the sensor vehicles are assumed to be assigned to specific subdomains, on which they are operating. The modified communication topology is presented in Subsection 4.3.1, whereas the slight modification of the observation model is subject of Subsection 4.3.2.

4.3.1 Communication Topology on Partitioned Domains

Again, a decentralized computing architecture without a central computing node is considered for the approach presented in this chapter. The communication topology is similar to the one described in Chapter 3, but provides some differences as groups of sensor vehicles are formed.

Each sensor vehicle is assigned to the subdomain it is located in. A group $N_i^{\Omega} = \{j : \mathbf{r}_j^{\mathbf{P}} \in \Omega_i\}$ of sensor vehicles is formed for each domain $i \in \{1, \ldots, n_d\}$. In the following, it is assumed that at least one vehicle is available per subdomain, i.e. $N_i^{\Omega} \neq \emptyset$. Between the members of the local group N_i^{Ω}, fast and efficient communication is possible.

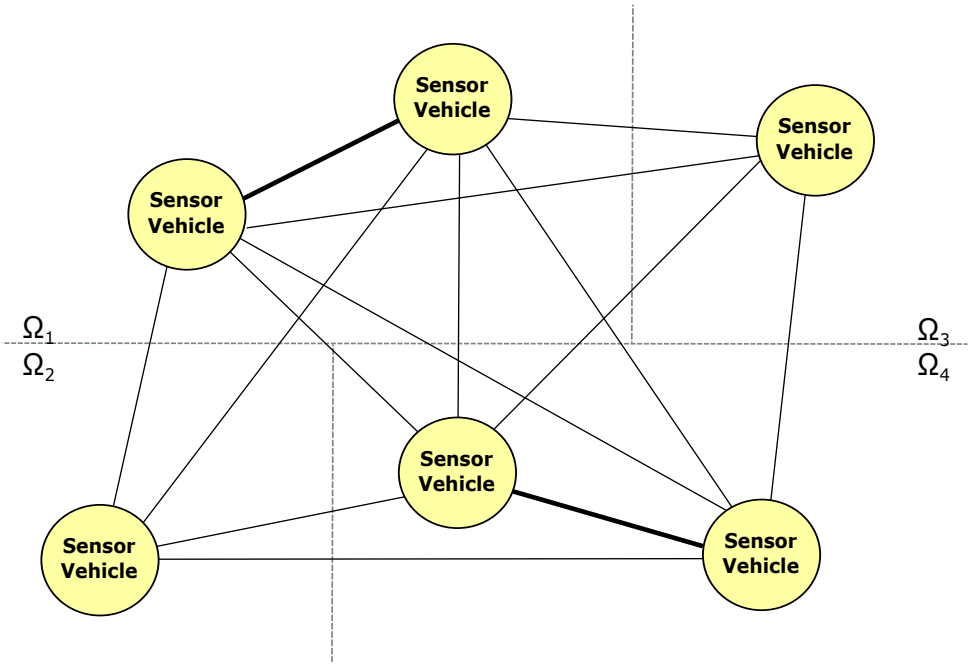


Figure 4.3: Communication network of the proposed partitioned monitoring approach. Each node communicates with nodes located in the same subdomain (bold link) and with nodes located on adjacent subdomains.

Furthermore, communication is also possible with groups $N_i^{\Gamma_{ij}} = N_j^{\Omega}$ operating on neighboring subdomains Ω_j. This means that communication, in total, is possible within the local group N_i^{Ω} as well as with all agents on neighboring subdomains $N_i^{\Gamma} = \cup_{j \in \{1,\ldots,n_d : \Gamma_{ij} \neq \emptyset\}} N_i^{\Gamma_{ij}}$. A direct communication with

all other sensor vehicles is not possible and information can be obtained only indirectly over neighbors. In this way, scalability is ensured and limited communication ranges are considered.

An overview over the resulting network topology is provided by Figure 4.3.

Every sensor vehicle maintains only the model of the subdomain it is assigned to. Assuming a perfect communication between the vehicles in a local group, the estimates of the group are identical. For global convergence of the monitoring approach, specific entities have to be exchanged between neighboring groups, especially regarding simulation (cf. Section 4.2) and data assimilation (cf. Section 4.4). Furthermore, it is possible for sensor vehicles to move from one subdomain to another (cf. Section 4.6) and to join another sensor group subsequently. Thus, the presented topology is not static but is likely to change over time.

4.3.2 Partitioned Observation Model

For the observation model considered in this chapter, a slight modification to (2.32) has to be made.

It is assumed that measurements can be passed quickly without effort between sensor vehicles located on the same subdomain. Therefore, it can be stated that the set of sensor vehicles N_i^{Ω} operating on the same subdomain Ω_i uses the same observation model

$$\mathbf{y}_i^{(k)} = \mathbf{H}_i^{(k)} \mathbf{x}^{\mathbf{t}(k)} + \boldsymbol{\epsilon}_i^{(k)}. \tag{4.7}$$

with observation vector $\mathbf{y}_i^{(k)} = \text{col}\{y_j^{(k)} : j \in N_i^{\Omega}\}$, observation model operator $\mathbf{H}_i^{(k)} = \text{col}\{\mathbf{h}_j^{(k)} : j \in N_i^{\Omega}\}$, model error $\boldsymbol{\epsilon}_i^{(k)} = \text{col}\{\epsilon_j^{(k)} : j \in N_i^{\Omega}\}$ and model error covariance matrix $\mathbf{R}_i^{(k)} = \text{blockdiag}\{R_j^{(k)} : j \in N_i^{\Omega}\}$.

4.4 Decentralized Data Assimilation on Partitioned Domains

Decomposing the domain into several subdomains with sensor vehicles working locally on them also requires a new approach to data assimilation for combining measurements and model forecasts. As described in Subsection 4.3.2, it is assumed that the local measurement vector $\mathbf{y}_i^{(k)}$ according to (4.7) is known at each vehicle for its subdomain Ω_i since exchanging the measurements within a sensor group is a lightweight task. Combining the forecast model with the measurement vector would result in the same analysis state vector for all sensor vehicles of the group. However, it is not sufficient to only consider the measurements of the sensors of the own subdomain for the analysis. Not considering measurements on neighboring subdomains can lead to discontinuities at the boundary between two subdomains leading to problems in the computation process. Furthermore, measurement information cannot be propagated in the opposite flow direction so that a neighboring subdomain located upstream of a measurement position at no time receives information about the measurement.

Therefore, not only the measurements of the own domain, but also measurements taken by sensors on neighboring domains should be considered. All other measurements that are available in subdomains not in the direct vicinity are not considered as they do not have a large impact on the state vector regarding the considered subdomain. This procedure improves scalability.

Using the measurements $\mathbf{y}_i^{(k)}$ as well as the measurements $\mathbf{y}_j^{(k)}$ from all adjacent subdomains for data assimilation results, however, not in a straightforward task. If the KF (see Section 2.4.2) is used, the combined error covariance matrix

$$\mathbf{P}_{i\vee j}^{(k)} = \begin{pmatrix} \mathbf{P}_i^{(k)} & \mathbf{P}_{ij}^{(k)} \\ \mathbf{P}_{ji}^{(k)} & \mathbf{P}_j^{(k)} \end{pmatrix} \tag{4.8}$$

must be known to account for the influence of $\mathbf{y}_j^{(k)}$ on $\mathbf{x}_i^{(k)}$. While the error covariance matrices $\mathbf{P}_i^{(k)}$ and $\mathbf{P}_j^{(k)}$ are known on their corresponding subdomain, the cross-covariance matrices $\mathbf{P}_{ij}^{(k)} = \left(\mathbf{P}_{ji}^{(k)}\right)^T$ are typically not. To apply the KF, the cross-covariance matrices would have to be additionally maintained, i.e. propagated in time and updated by measurements. This requires either a huge communication effort or the knowledge of the model operator of every neighboring domain. For these reasons, the KF approach has to be rejected for the scenarios considered in this chapter.

A much more efficient approach is obtained when working with an ensemble variant. Therefore, an ensemble-based data assimilation method for partitioned systems is proposed in this section.

This approach does not only save computation time by representing the error covariance matrix implicitly (see Section 2.4.3), but also allows an efficient and lightweight exchange of measurement and uncertainty information between different subdomains. A further advantage of the approach is its implicit consideration of localization since distant measurements do no have an impact. Thus, the proposed assimilation approach is closely related to local ensemble filters [132, 83] as well as to parallel ensemble filters [98, 81], but is motivated by the task of decentralized estimation and treats forecast *and* analysis locally.

The data assimilation strategy is presented in the following starting with the ensemble representation and the explanation of the forecast step in Subsection 4.4.1 followed by the description of the update step in Subsection 4.4.2.

4.4.1 Ensemble Representation and Forecast

The ensemble that is used for the partitioned estimation is set up on the global domain Ω as introduced in Section 2.4.3. Having constructed the ensemble members, the different state vectors are decomposed to obtain the local ensemble on the subdomains

$$\mathbf{x}_{i(l)}^{(0)} = \mathrm{col}\{(\mathbf{x}_{(l)}^{(0)})_j : \mathbf{r}_j \in \Omega_i\}. \tag{4.9}$$

This procedure guarantees that the ensemble members are continuous at the boundaries Γ_{ij} between the subdomains with corresponding ensemble members $\mathbf{x}_{i(l)}^{(k)}$ and $\mathbf{x}_{j(l)}^{(k)}$.

In this way, every ensemble member can be forecasted according to Section 4.2. The boundary vectors $\mathbf{x}_{i(l)}^{\Gamma_{ij}(k)}$ of every ensemble member are sent from subdomain Ω_i to the downstream subdomain Ω_j, where the received boundary vector is used as Dirichlet condition for the corresponding ensemble member $\mathbf{x}_{j(l)}^{\Gamma_{ji}(k)} = \mathbf{x}_{i(l)}^{\Gamma_{ij}(k)}$. For every boundary, this procedure requires an exchange of an $n_{\Gamma_{ij}} \times n_s$-dimensional matrix where $n_{\Gamma_{ij}}$ is the size of the boundary vector $\mathbf{x}_i^{\Gamma_{ij}(k)}$.

This forecast procedure results not only in the mean estimate being able to evolve across subdomain boundaries, also the error covariance matrix, implicitly represented by the ensemble spread, is propagated from one domain to the other.

4.4.2 Efficient Analysis Step

In the analysis step, the forecasted ensemble of every subdomain is combined with the measurements available in the subdomain as well as with the measurements of the neighboring subdomains. Using the EnSRF (cf. Subsection 2.4.3), the decentralized analysis can be achieved in an efficient manner with a minimal exchange between the sensor vehicles on different subdomains.

In the following, all entities relate to the same time stage $t = t^{(k)}$ so that the superscript $(\cdot)^{(k)}$ is omitted for reasons of clarity.

According to (2.45), the Kalman gain $\mathbf{K}$ plays a decisive role when updating the mean state vector. Defining the projected error covariance $\mathbf{Z} = \mathbf{H}\mathbf{X}^{\mathbf{f}} \in \mathbb{R}^{n_p \times n_s}$, the global Kalman gain $\mathbf{K}$ can be expressed as

$$\mathbf{K} = \mathbf{X}^{\mathbf{f}}\mathbf{Z}^T \left(\mathbf{Z}\mathbf{Z}^T + \mathbf{R}\right)^{-1} \tag{4.10}$$

so that the composed Kalman gain $\tilde{\mathbf{K}}$ corresponding to the composed state vector in (4.6) reads

$$\tilde{\mathbf{K}} = \begin{pmatrix} \mathbf{K}_1 \\ \vdots \\ \mathbf{K}_i \\ \vdots \\ \mathbf{K}_{n_d} \end{pmatrix} = \begin{pmatrix} \mathbf{X}_1^{\mathbf{f}} \\ \vdots \\ \mathbf{X}_i^{\mathbf{f}} \\ \vdots \\ \mathbf{X}_{n_d}^{\mathbf{f}} \end{pmatrix} \cdot \mathbf{F}(\tilde{\mathbf{Z}}, \mathbf{R}) \tag{4.11}$$

with a matrix $\mathbf{F}$ depending on the composed projected error covariance $\tilde{\mathbf{Z}} = \text{col}\{\mathbf{Z}_i\}_{i=1}^{n_d}$ and the global error covariance matrix $\mathbf{R}$. Relation (4.11) shows that, to generate the local Kalman gain $\mathbf{K}_i$, only the local ensemble spread $\mathbf{X}_i^{\mathbf{f}}$, the error covariance at the observation site $\tilde{\mathbf{Z}}$ and the error covariance matrix $\mathbf{R}$ must be known.

It is assumed that measurements on subdomains Ω_j that are not adjacent to subdomain Ω_i do not have an impact on the analysis $\mathbf{x}_i^{\mathbf{a}}$. This means that $\mathbf{P}_{ij}^{\mathbf{f}} = \mathbf{X}_i^{\mathbf{f}} \left(\mathbf{X}_j^{\mathbf{f}}\right)^T = \mathbf{Z}_i\mathbf{Z}_j^T = 0$ and $\mathbf{Z}_j$ is not needed to generate the local Kalman gain $\mathbf{K}_i$. Instead, following (4.11), the local Kalman gain yields

$$\mathbf{K}_i = \mathbf{X}_i^{\mathbf{f}}(\mathbf{Z}_i^{\mathbf{c}})^T \left(\mathbf{Z}_i^{\mathbf{c}}(\mathbf{Z}_i^{\mathbf{c}})^T + \mathbf{R}_i^{\mathbf{c}}\right)^{-1}. \tag{4.12}$$

with $\mathbf{Z}_i^{\mathbf{c}} = \text{col}\{\mathbf{Z}_j : j = i \vee \Gamma_{ij} \neq \emptyset\}$ and $\mathbf{R}_i^{\mathbf{c}} = \text{blockdiag}\{\mathbf{R}_j : j = i \vee \Gamma_{ij} \neq \emptyset\}$. Regarding communication, only low-dimensional entities have to be exchanged to generate $\mathbf{K}_i$ with information from subdomain Ω_j: the $n_{p_j} \times n_s$ matrix $\mathbf{Z}_j$ and the observation error variance $\mathbf{R}_j$ with n_{p_j} diagonal elements where n_{p_j} is the number of measurements on subdomain Ω_j.

Furthermore, the n_{p_j}-dimensional innovation vector $\mathbf{d}_j$ has to be exchanged so that the ensemble mean in every subdomain can be updated according to (2.36) with

$$\bar{\mathbf{x}}_i^{\mathbf{a}} = \bar{\mathbf{x}}_i^{\mathbf{f}} + \mathbf{K}_i\mathbf{d}_i^{\mathbf{c}}, \tag{4.13}$$

where $\mathbf{d}_i^{\mathbf{c}} = \text{col}\{\mathbf{d}_j : j = i \vee \Gamma_{ij} \neq \emptyset\}$.

The ensemble spread is updated using the transformation matrix $\mathbf{T}$ as in (2.46) According to (2.47), the transformation matrix only depends on $\mathbf{R}$ and $\mathbf{Z}$ so that no further entities are necessary to update the ensemble

$$\mathbf{T}_i\mathbf{T}_i^T = \mathbb{1} - (\mathbf{Z}_i^{\mathbf{c}})^T(\mathbf{Z}_i^{\mathbf{c}}(\mathbf{Z}_i^{\mathbf{c}})^T + \mathbf{R}_i^{\mathbf{c}})^{-1}\mathbf{Z}_i^{\mathbf{c}} \tag{4.14}$$

$$\mathbf{X}_i^{\mathbf{a}} = \mathbf{X}_i^{\mathbf{f}}\mathbf{T}_i \tag{4.15}$$

$$\mathbf{x}_{i(l)}^{\mathbf{a}} = \bar{\mathbf{x}}_i^{\mathbf{a}} + (\mathbf{X}_i^{\mathbf{a}})_{\cdot,l}. \tag{4.16}$$

A summary of the partitioned forecast and update procedure is provided by Algorithm 2 showing the procedure each vehicle is performing in every step.

Data: ensemble set $\{\mathbf{x}_{i(l)}^{(k-1)}\}_{l=1}^{n_s}$ at old time $t = t^{(k-1)}$
Result: updated ensemble set $\{\mathbf{x}_{i(l)}^{\mathbf{a}(k)}\}_{l=1}^{n_s}$ at new time $t = t^{(k)}$
for $j \in \{1, \ldots, n_d : \Gamma_{ij}^{out} \neq \emptyset\}$ **do**
  send $\{\mathbf{x}_{i(l)}^{\Gamma_{ij}(k-1)}\}_{l=1}^{n_s}$ to neighbors N_j^{Ω} operating on subdomain Ω_j;
end
for $j \in \{1, \ldots, n_d : \Gamma_{ij}^{in} \neq \emptyset\}$ **do**
  receive $\{\mathbf{x}_{i(l)}^{\Gamma_{ij}(k-1)}\}_{l=1}^{n_s}$ from neighbors N_j^{Ω} operating on subdomain Ω_j;
  set received data as Dirichlet condition on Γ_{ij}^{in};
end
forecast ensemble set $\{\mathbf{x}_{i(l)}^{(k-1)}\}_{l=1}^{n_s}$ to new time $t^{(k)}$;
if *new measurement time* $t^{(k)}$ *reached* **then**
  exchange $y_m^{(k)}$,$R_m^{(k)}$,$\mathbf{r}_m^{\mathbf{p}(k)}$ with local neighbors N_i^{Ω};
  assemble and compute $\mathbf{y}_i^{(k)}$,$\mathbf{d}_i^{(k)}$, $\mathbf{H}_i^{(k)}$, $\mathbf{Z}_i^{(k)}$, $\mathbf{R}_i^{(k)}$;
  exchange $\mathbf{d}_i^{(k)}$, $\mathbf{Z}_i^{(k)}$, $\mathbf{R}_i^{(k)}$ with sensor vehicles N_i^{Γ} on neighboring subdomains;
  assemble $\mathbf{Z}_i^{\mathbf{c}(k)}$, $\mathbf{R}_i^{\mathbf{c}(k)}$;
  perform update procedure (4.13)-(4.16);
end

Algorithm 2: Proposed forecast and update procedure of the partitioned monitoring approach for a sensor vehicle $m \in \{1, \ldots, n_V\}$ operating on domain Ω_i

It should be noted that, if the considered subdomains are still large, covariance localization (2.50) can be applied additionally to prohibit unphysical correlations between distant locations. In this case, the positions $\mathbf{r}^{\mathbf{p}}$, at which the considered measurements are taken, should be also exchanged so that the localization matrices $\mathbf{\Psi}_1$ and $\mathbf{\Psi}_2$ can be computed.

4.5 Identification of Measurement Locations on Partitioned Domains

Based on the estimates and the statistics of the estimates, suitable measurement locations have to be identified as input for the vehicle controller. The strategy used for the decentralized, partitioned approach is mainly based on the strategy proposed in Sections 2.5 and 3.5 for centralized and decentralized systems on a full domain, but is modified to be applicable for partitioned domains. Moreover, an extension allowing the sensor vehicles to move from one domain to another is established.

Locations belonging to a high error variance and probable source locations are, still, suitable positions at which measurements should be taken. This is why the target point identification procedure as described before can be also applied in the scenario considered here, with the difference that only the subdomain the sensor vehicles are located on is considered. Thus, the error covariance matrix $\mathbf{P}_i^{(k)}$ (or the ensemble spread $\mathbf{X}_i^{(k)}$, respectively) is used to identify suitable measurement locations on the subdomain i instead of using the global error covariance matrix $\mathbf{P}^{(k)}$. The possible source target points are determined in the same manner, considering only terms that are locally available. Target points are calculated at each sensor vehicle separately. As the estimates and models of sensors on a common subdomain are identical, the group of sensor vehicles on a common subdomain shares an identical set of target points enabling a consistent cooperation. The number of target points is chosen to correspond to the number of sensor vehicles on the domain. Local appealing points (see Subsection 2.5.2) can be calculated for every sensor vehicle to reduce uncertainty on the way to the target locations. Hence, the basis of the identification procedure for the partitioned approach resembles the centralized approach performed on several subdomains.

Assigning sensor vehicles to fixed subdomains limits the adaptivity of the whole monitoring approach. The considered applications involve a dynamic process and so it might be ineffective to work with a fixed distribution of sensors to subdomains. From a sensor's perspective, a measurement in an adjacent subdomain could be much more suitable than in the own subdomain, e.g. when the uncertainty in the neighboring subdomain is extremely high, while the estimates in the own subdomain are rather good. Therefore, an adaptive method, in which, under specific conditions, the sensor vehicles are allowed to move from one subdomain to another, should be established.

Introducing a third type of target point can alleviate this problem. From time to time, the total subdomain variance $V_i = \text{trace}(\tilde{\mathbf{P}}_i) = \sum_{j=1}^{n_i} \sum_{l=1}^{n_s} (\mathbf{X}_i)_{j,l}^2$ is exchanged between neighbors. If the condition

$$\frac{V_i}{n_{p,i} - 1} < \frac{V_j}{n_{p,j} + 1}, \tag{4.17}$$

holds for subdomains Ω_i and Ω_j, a sensor vehicle of subdomain Ω_i should move to subdomain Ω_j. In that case, the last target point should be replaced by a target point on the boundary Γ_{ij}. As soon as this target point is reached by a sensor vehicle from domain Ω_i, it transitions to domain Ω_j and is supplied with the necessary domain information (state ensemble, forecast model, etc) by the group of domain Ω_j. Obviously, the communication effort needed for the initialization of the new team member is significant, but can be accepted since a transition to a new domain is not performed in a high frequency. Furthermore, a maximum number of sensor vehicles per subdomain is defined to prevent all sensor vehicles from driving to the same subdomain and to preserve scalability with respect to the number of sensor vehicles.

4.6 Vehicle Controller for Application on Partitioned Domain

The final part of the partitioned monitoring approach is to guide the sensor vehicles of every subdomain to their corresponding target locations. Again, it would be beneficial, if a cooperative, optimization-based vehicle controller with feedback control is applied. It turns out that the decentralized vehicle controller of Section 3.6 can be used in a straightforward manner for the decentralized, partitioned scenario. Every vehicle computes its own control input by considering optimization problem (3.22) with respect to its local subdomain Ω_i. To account for the movement of the other sensor vehicles on the subdomain and to

achieve cooperation, the local state vector $\tilde{\mathbf{p}}_i^{(k)}$ is shared by all vehicles of subdomain Ω_i and comprises the vehicle state $\mathbf{p}_j^{(k)}$ as well as the target point state $\mathbf{q}_j^{(k)}$ of all vehicles and targets $j \in N_i^{\Omega}$. As a result, the sequence of control inputs $U_{n_{Tr},i}^{(k)}$ is obtained for all vehicles of the same subdomain, but only the first input concerning the own sensor vehicle is actually applied. The other vehicles on the subdomain employ the same procedure so that every vehicle applies its optimal input resulting in a cooperative movement regarding the task of getting to the target locations. The procedure is repeated in the following time steps with potentially new target points where the optimization problem is considered over a shifted time horizon according to MPC.

If a sensor vehicle reaches a boundary target point, it leaves its current subdomain and becomes a member of the vehicle group on the neighboring subdomain. The communication topology has to be updated and the transition procedure described in Section 4.5 is applied.

4.7 Implementation, Evaluation and Results

In this section, the performance of the proposed partitioned monitoring strategy is evaluated.

Subsection 4.7.1 provides a few remarks on the implementation of the partitioned approach. The general working of the proposed partitioned approach is shown in Subsection 4.7.2 and comparisons with a centralized monitoring approach on a full domain, i.e. the centralized monitoring approach proposed in Chapter 2 are drawn. Afterward, the influence of the partitioned simulation and partitioned data assimilation procedure are studied in Subsection 4.7.3 and aspects related to computation time for a different number of grid points, subdomains, and sensor vehicles are addressed in Subsection 4.7.4.

4.7.1 Implementation

The MATLAB code for the centralized DDDAS (see Subsection 2.7.1) and decentralized DDDAS (see Subsection 3.7.1) has been further extended to enable the simulation of the proposed partitioned dynamic data-driven monitoring approach. The extension includes, amongst others, a modification of the meshing procedure to obtain suitable partitioned computation grids and the implementation of the required domain decomposition as well as the proposed data assimilation methods. Furthermore, vehicle control and target generation have to be adapted to account for the changed requirements. Again, the communication topology is likely to change so that several possible structures for the vehicle controller have to be precalculated.

4.7.2 General Results of the Proposed Partitioned Approach

The partitioned approach is especially designed for larger problem domains so that the test case domain $\Omega = [0\,\mathrm{m}, 4000\,\mathrm{m}] \times [0\,\mathrm{m}, 4000\,\mathrm{m}]$ is chosen to be much larger than the one considered in the previous chapters. For the test case scenario, the process velocity is constant and amounts to $w_1 = 3\,\mathrm{ms}^{-1}$ in r_1-direction and $w_2 = 5\,\mathrm{ms}^{-1}$ in r_2-direction, whereas the diffusivity amounts to $D = 5\,\mathrm{m}^2\mathrm{s}^{-1}$. With these choices, the problem can be considered to be advection-dominated and d-APN can be applied concerning process simulation. The true initial concentration is composed of five Gaussian pulses as depicted in Figure 4.5(b). Two sources are considered yielding the source term

$$s(\mathbf{r}, t) = 0.005 f(\mathbf{r} - \mathbf{r}_{\mathbf{s},1}, 100) + 0.008 f(\mathbf{r} - \mathbf{r}_{\mathbf{s},2}, 200) \tag{4.18}$$

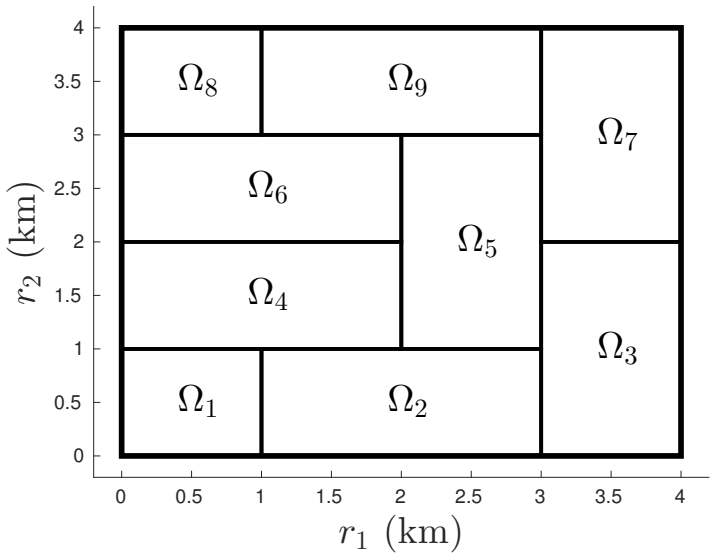


Figure 4.4: Decomposition of the problem domain used in the test case scenarios

with $\mathbf{r}_{\mathbf{s},1} = (250\,\mathrm{m}, 450\,\mathrm{m})^T$ and $\mathbf{r}_{\mathbf{s},2} = (500\,\mathrm{m}, 350\,\mathrm{m})^T$ and local compact support function f from [63].

While the true sources are assumed to be known, model error consisting of perturbations of the velocity and diffusivity of the true state and true initial state are not. By contrast, the initial estimate is set to zero.

The domain is decomposed into nine subdomains depicted in Figure 4.4. As stated in Section 4.3, it is assumed that only measurements on adjacent subdomains influence the local state estimate. Thus, the decomposition has been designed such that distances between non-adjacent subdomains are large. The computation grid of the full problem domain consists of approximately $n = 2,000$ nodes leading to subdomain grid sizes between 100 and 300 nodes.

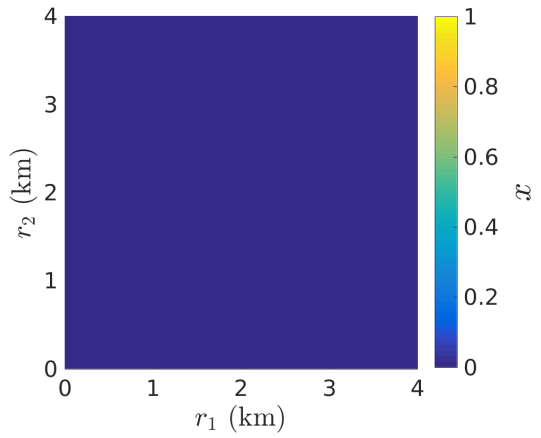
In total, 18 sensor vehicles are applied in the monitoring scenario – two in every domain.

The partitioned monitoring strategy described in this chapter is applied with the settings listed in Table 4.1 and a simulation is conducted over $400\,\mathrm{s}$ resulting in $n_T = 200$ time steps.

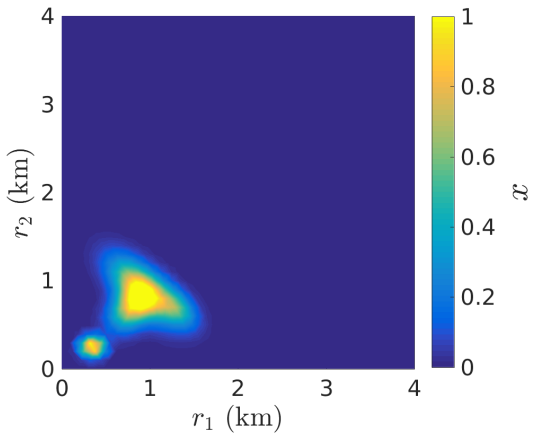
The results obtained with the partitioned monitoring strategy are depicted in Figure 4.5. Compared to the true solution at the same time, the behavior, which could already be observed in the previous chapter, also shows here. A rather bad estimate at the initial time is improved over time and qualitative differences are decreasing with increasing monitoring time. At final time, a rather good estimate of the true process state becomes available.

While the qualitative results look promising, the error of the partitioned approach has to be investigated quantitatively. In particular, a comparison between the new decentralized approach operating on a partitioned domain and a centralized approach working on the full domain is interesting. Therefore, the centralized approach presented in Chapter 2 is also applied in the described test case scenario.

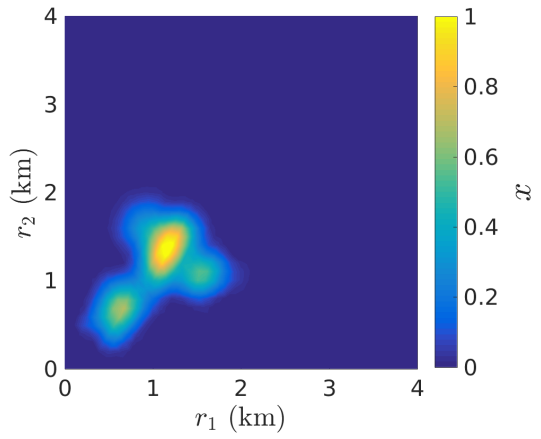
However, the vehicle control step of the centralized monitoring approach with 18 sensor vehicles has such a high complexity that required computation times for one step partially exceed one hour or even lead to a breakdown of the simulation. Therefore, the MPC horizon for the centralized approach has to be set to $n_{Tr} = 10$, resulting in a poorer quality of the generated trajectories.



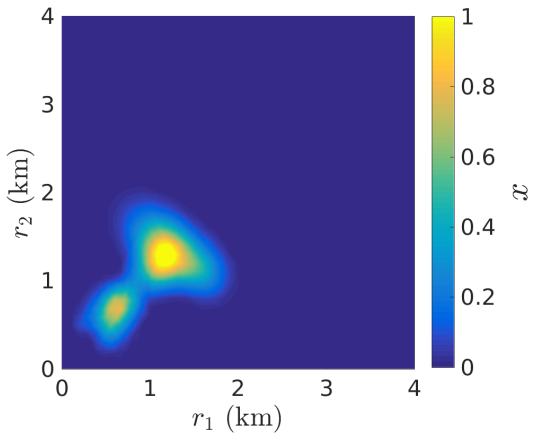
(a) Estimated concentration at $t = 0s$



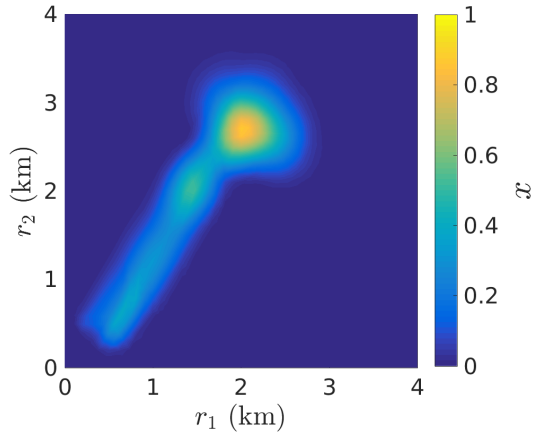
(b) True concentration at $t = 0s$



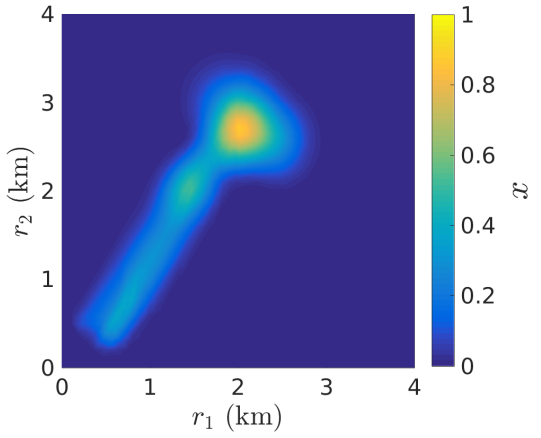
(c) Estimated concentration at $t = 100s$



(d) True concentration at $t = 100s$



(e) Estimated concentration at $t = 400s$



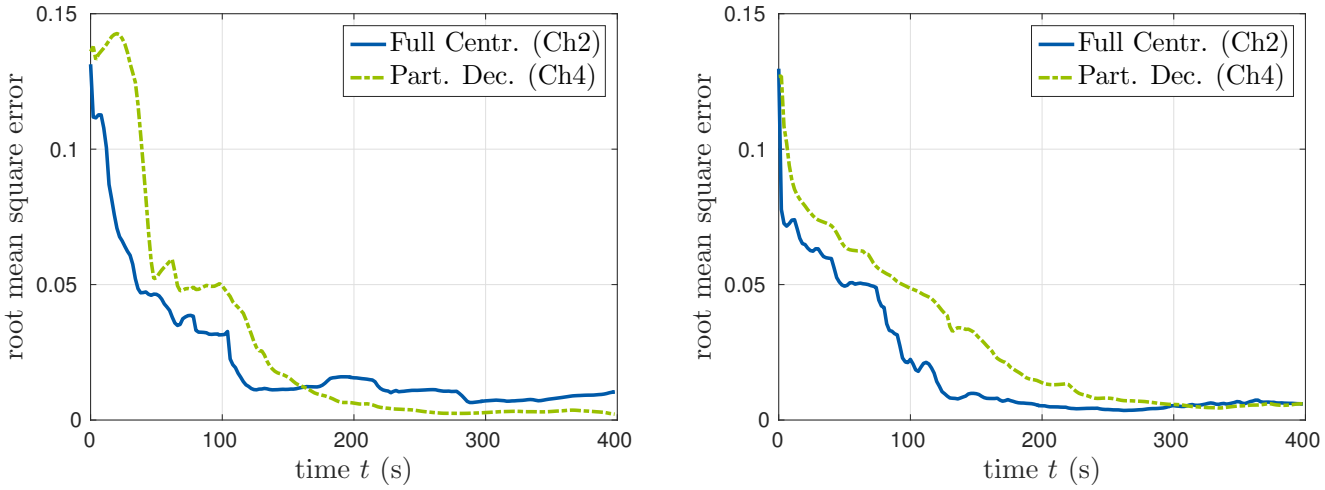
(f) True concentration at $t = 400s$

Figure 4.5: Concentration distribution at different times – estimate (left) vs. true state (right)

Table 4.1: Data assimilation and controller settings

Ensemble size n_s	20
Observation variance factor R_A	$2 \cdot 10^{-2}$
Minimum observation variance R_{min}	10^{-3}
Initial covariance magnitude* P_A	$0.8, 0.5$
Initial covariance center* $\mathbf{r_P}$ [m]	$(900, 900)^T$, $(350, 350)^T$
Initial covariance area* L_{P1} [m^2]	$1.3 \cdot 10^5$, $2 \cdot 10^4$
Initial covariance area* L_{P2} [m^2]	$2 \cdot 10^5$
Model covariance magnitude Q_A	$2 \cdot 10^{-2}$
Model covariance area L_Q [m^2]	$2 \cdot 10^5$
Localization function parameter c_G [m]	300
Maximum vehicle speed v^p_{max} [ms^{-1}]	10
Maximum vehicle acceleration a^p_{max} [ms^{-2}]	10
MPC horizon** n_{Tr}	10, 20
Number of hyperplanes n_γ	8

* $\mathbf{P}^{(0)}$ consists of the linear combination of two initial error covariances according to (2.87)
**referring to two different simulation runs



(a) $n_d = 9$, $n_{Tr} = 20$ (dec.), $n_{Tr} = 10$ (centr.) (b) $n_d = 4$, $n_{Tr} = 20$

Figure 4.6: Root mean square error of the state estimate obtained with the proposed centralized monitoring approach in Chapter 2 on full domain compared to the error of the proposed partitioned decentralized monitoring approach of Chapter 4 over time

Figure 4.6 provides the resulting error in state estimate for the two approaches. Both approaches lead to a significant reduction of the mean error. In the centralized approach, the error is decreased rapidly before it is kept constant. The partitioned approach shows the same behavior, but requires a longer time to reduce the error. The reason for this is that the partitioned approach is less flexible than the centralized approach acting on the full domain. Sensors have to be redistributed first to attain an assignment to subdomains capable of adequately monitoring the dispersion process. Thus, the decentralized partitioned approach shows the same delay behavior as the decentralized approach with limited communication ranges (cf. Figure 3.5). At final time, the estimation quality of the partitioned approach is even better than the one obtained with the centralized approach. This issue is related to the reduced MPC horizon $n_{Tr} = 10$ of the centralized approach leading to sensor vehicle trajectories with a worse quality. On the other side, the partitioned approach uses a horizon of $n_{Tr} = 20$ so that vehicle movement is planned over a longer time horizon.

To compare the behavior of the centralized and the partitioned approach with a common MPC horizon, a scenario with less sensor vehicles is considered. In this scenario, the domain is decomposed into four subdomains as indicated in Figure 4.4 with, again, two sensor vehicles in each domain. The resulting state estimation error obtained with an MPC horizon of $n_{Tr} = 10$ is shown in Figure 4.6(b) and can be also characterized by a delay in error reduction. However, the estimation error for both approaches is almost equal at final time.

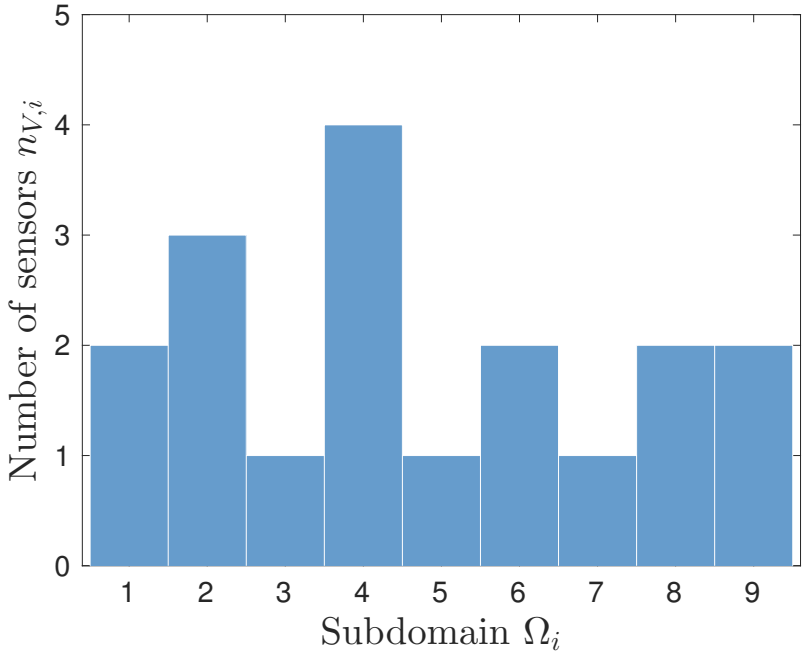


Figure 4.7: Histogram of sensor distribution on subdomains Ω_i at time $t = 100\,\mathrm{s}$

Investigating the behavior and movement of the sensor vehicles, it has to be checked if the sensor network practically succeeds in adaptively distributing over the subdomains. The histogram in Figure 4.7 shows the number of sensor vehicles in each subdomain at time $t = 100\,\mathrm{s}$. The estimate and true state corresponding to this time stage are depicted in Figure 4.5(c) and 4.5(d). Most of the concentration to be estimated is located in the Subdomains Ω_1, Ω_2 and Ω_4, which have gained in total three sensor vehicles. As the area of Subdomain Ω_1 is small compared to those of the other subdomains, two sensor vehicles are sufficient and no additional sensor vehicle is moved to Ω_1. On the other side, less uncertainty is

expected in the other subdomains so that these subdomains have ceded three sensor vehicles in total. Hence, a dynamical redistribution has taken place.

4.7.3 Investigation of the Influence of Partitioning

Forecasting the process state requires, besides the solution of the local subproblem, the exchange of boundary data between neighboring subdomains to obtain a globally convergent solution. An efficient and decoupled exchange can be applied if d-ADN is used. The values at the boundary nodes are passed as Dirichlet conditions to the downstream subdomains and homogeneous Neumann conditions are set at outflow boundaries. This strong simplification, however, only works if the process is dominated by advection.

In order to assess for which scenarios the application of d-ADN is justified, the forecast part of the problem described in the previous subsection is performed with different diffusion parameters D. The error between the solution on the full grid and the solution on the partitioned grid is calculated and normalized by the total amount of concentration on the full grid to account for concentration losses due to diffusion out of the problem domain. Figure 4.8 depicts the obtained relative errors for different diffusivities. Clearly, the partitioned forecast method is not suitable for large diffusivities ($D = 5000\,\mathrm{m^2 s^{-1}}$) leading to a large growth of error. However, for small diffusion coefficients, which have been considered in this work, the error is increasing only slightly over time. Choosing $D = 5\,\mathrm{m^2 s^{-1}}$, the resulting relative error is still under 10% after 200 time steps. This is, of course, not an excellent result concerning highly accurate forecast methods, but applying a mobile sensor network with a data fusion method explicitly considering model error, the forecast error can be compensated and the proposed method can be accepted.

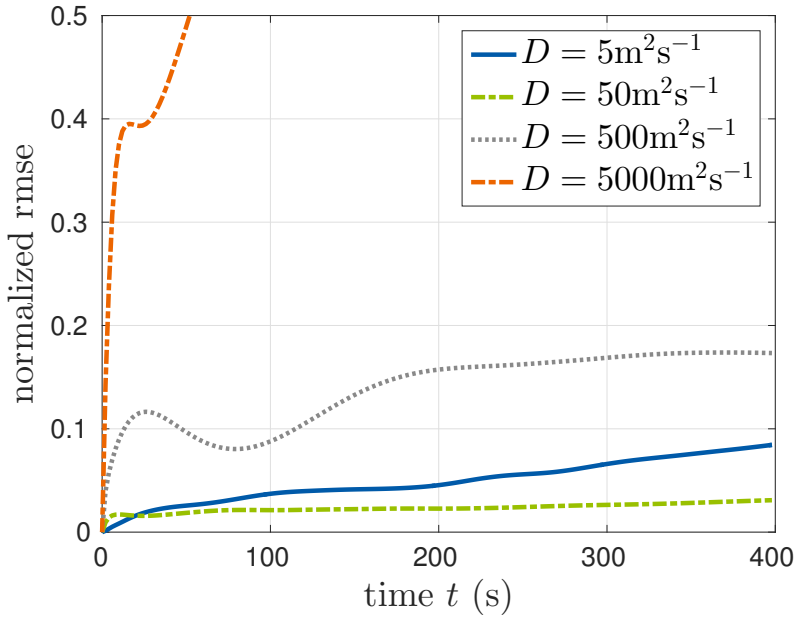


Figure 4.8: Relative error between simulation results on full domain and simulation results on decomposed domain with d-ADN for different diffusivities D over time

Provided that only measurements on neighboring domains have an influence on local partitioned state estimates, the partitioned analysis procedure does not insert error to the system when compared to the analysis on the full domain. Assuming a localization radius of $300\,\mathrm{m}$, all correlations over a range of

more than $600\,\mathrm{m}$ are cut off so that the application of the partitioned analysis scheme for the domain decomposition shown in Figure 4.4 actually results in no additional error.

However, the partitioned data assimilation approach is based on an ensemble set that is propagated using the partitioned forecast step, which introduces error as investigated before. For this reason, the error reduction made with partitioned data assimilation is investigated and compared to its full domain variant. A static sensor network with 18 sensors is considered to exclude the influence of mobile sensor vehicles.

Figure 4.9 shows the resulting estimation errors for the data assimilation method on the full domain and for the data assimilation method on the partitioned domain. The partitioned state forecast provokes a slightly worse estimate at the beginning, but, due to measurement updates, this additional error is compensated and both approaches show a comparable error reduction over time.

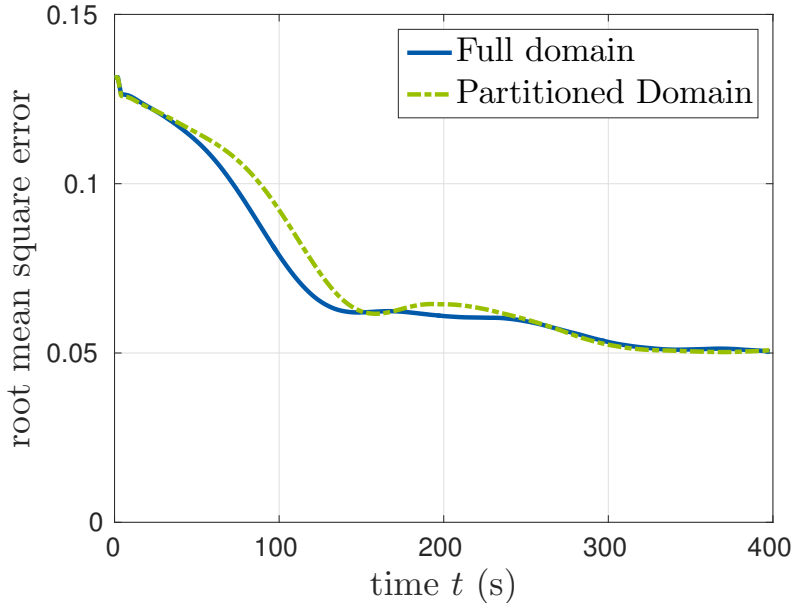


Figure 4.9: Root mean square error of the state estimate obtained with data assimilation and simulation on decomposed domain vs. error of the state estimate obtained with data assimilaton and simulation on global domain

4.7.4 Computation Time and Scalability

Finally, the required computation time and scalability are investigated for the proposed partitioned monitoring approach following the procedure described in Subsections 2.7.5 and 3.7.5.

Figure 4.10 provides the resulting times for one forecast and update step for different numbers of the original domain n and for different decompositions resulting in a different number of subdomains n_d. The results obtained on the full domain ($n_d = 1$) are comparable to the results obtained in Subsection 2.7.5 permitting a comparison between the presented approaches in Chapters 2 and 4. As expected, the computation time decreases with increasing number of subdomains. For nine subdomains, a significant reduction of computing time is achieved. However, domain decomposition considering a fixed number of subdomains is not as scalable as the use of reduced order models and it is recommended to increase the number of subdomains n_d with increasing n. This is particularly suitable if the considered

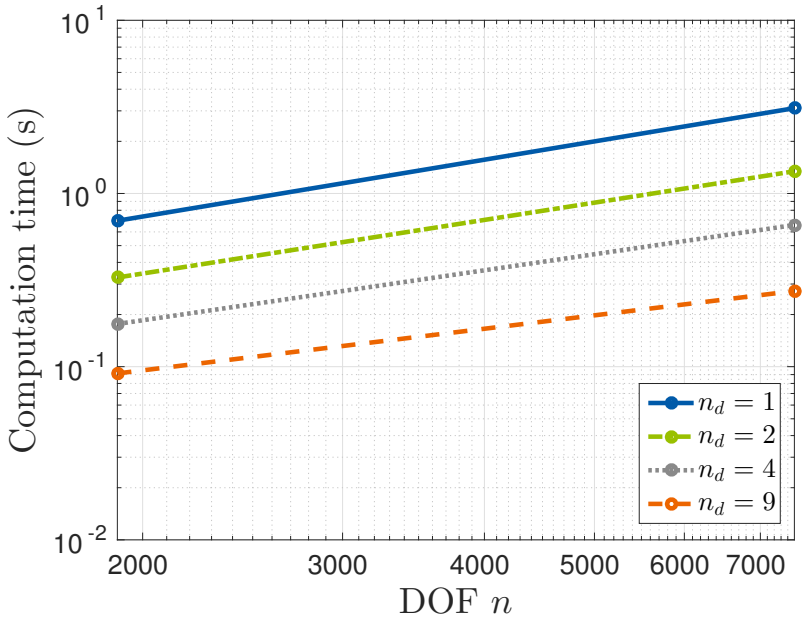


Figure 4.10: Average computation time for one forecast and assimilation step of the proposed partitioned monitoring approach in dependence of the original problem dimension n and the number of subdomains n_d

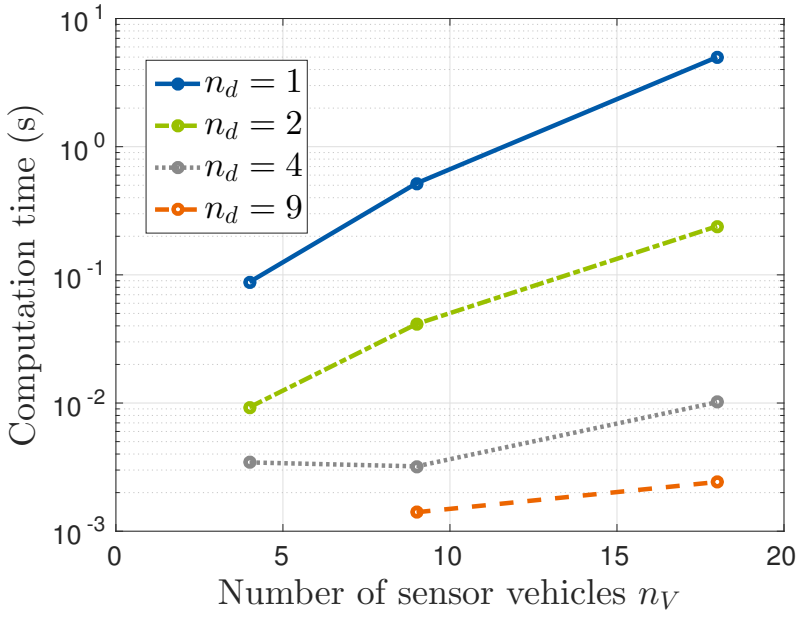


Figure 4.11: Average computation time for one vehicle control step of the proposed partitioned monitoring approach in dependence of the number of applied sensor vehicles n_V and the number of subdomains n_d

problem domain has large spatial dimensions. If the spatial dimensions are low, but the number of grid points is high, reduced order models and the decentralized approach of Chapter 3 should be preferred.

The computing times for one step of the vehicle control part are shown in Figure 4.11, which relies on results obtained from simulations with $n_{Tr} = 10$ so that a direct comparison with Figures 2.27 and 3.12 is not possible. However, the setting $n_d = 1$ leads to a monitoring approach very similar to the centralized approach proposed in Chapter 2 so that a comparison between the centralized and the partitioned approach becomes possible. Clearly, the computing time is reduced drastically with an increasing number of subdomains. Setting $n_{Tr} = 20$, the computing time difference between both approaches becomes even more amplified. While the centralized approach produces computing times of over an hour per step and leads to a breakdown of the simulation, the decentralized partitioned approach, on average, requires only 0.1 s with $n_d = 9$. The reason for this is that less vehicles have to be considered in the vehicle control problem so that the problem can be solved much faster. Compared to the decentralized monitoring approach presented in Chapter 3, a further great benefit of the partitioned method is the possibility to declare a maximum number of sensor vehicles per subdomain so that the number of vehicles to be considered in the local vehicle control problem is limited. Thus, the partitioned approach can be considered to be more scalable with respect to a higher number of sensor vehicles, at the cost of less flexibility and, hence, a less estimation quality.

4.8 Summary and Discussion of Chapter 4

Another new decentralized PDE-based dynamic data-driven monitoring strategy for atmospheric dispersion processes has been presented in this chapter. The central idea of the approach is the decomposition of the global problem domain into smaller subdomains. Sensor vehicles are assigned to subdomains and only have to maintain a model of their local subdomain instead of one of the global domain. Furthermore, only vehicles within the common local subdomain have to be involved in the vehicle control problem. In this way, scalability, both with respect to a larger problem domain and with respect to an increased number of sensor vehicles can be attained.

The challenge in this scenario is to guarantee a global convergence by solving local subproblems. Important information have to be communicated between agents residing on neighboring domains. While the basic structure of the monitoring approach of the previous chapters has been kept, a new prediction and update procedure based on d-ADN and EnSRF has been proposed that requires only little exchange between vehicles on neighboring domains. Furthermore, an extension of the target-point-based vehicle controller has been developed that allows a dynamic redistribution of sensor vehicles to account for a dynamically changing process.

Test cases have shown a better scalability of the present approach compared to the centralized method but also a slightly higher estimation error, especially at the beginning of the application. Sensor vehicles in the partitioned approach are, despite the dynamic redistribution property, not as flexible as in approaches operating on global domains so that trajectories cannot be considered to be globally optimal. It has been further shown that an additional model error is inserted to the system if the chosen state propagation method is used. While the influence of this error can be reduced via data assimilation, a small region should not be partitioned into too many subdomains. Therefore, the decentralized approach is especially suitable for large spatial domains.

For future work, the inclusion of reduced models should be considered making the proposed approach also scalable with respect to the size of the subdomain. A starting point could be found, for example, in [12].

5 Conclusion

Three new approaches to adaptively monitor atmospheric dispersion processes with the aim of process state and parameter estimation have been proposed in this work. All approaches are based on the predictions of a discretized PDE model enabling physically realistic forecasts. Furthermore, the approaches are dynamic data-driven: Mobile sensor vehicles are employed whose control inputs are determined online based on current process estimates. In this way, sensor vehicles can adapt their positions so that measurements are gathered at locations providing a higher information gain. Consequently, higher quality estimates of the atmospheric dispersion process can be obtained online.

The monitoring approaches further support cooperation, i.e. efficient work sharing among sensor vehicles and, at least partially, scalability with respect to larger problem domains and a higher number of vehicles. A key contribution in this context is the development of *decentralized* PDE-based dynamic data-driven monitoring approaches. The proposed decentralized approaches not only increase scalability and robustness, but also consider limited communication capabilities, especially limited communication ranges.

Designing approaches with the stated properties is challenging since, in practice, the solutions of high-dimensional problems, typically associated with PDE models, have to be available in a very short amount of time. To face this challenge, the proposed procedural structure avoids the solution of a complex optimal control problem and splits the complete problem into several building blocks consisting of process simulation, data fusion, target point generation and vehicle control. The individual building blocks have to be developed regarding the required computation time aspect as well as the demand for a good quality estimation. Furthermore, considering decentralized approaches, the individual methods have to account for only local availability of information and potentially different estimates at different vehicles.

The first monitoring approach, proposed in **Chapter 2**, is based on a centralized communication network with a central computing machine, which is responsible for the computational work, receives measurements from the sensor vehicles and sends control inputs back to them. To account for the efficiency aspect, besides the application of the mentioned procedural structure, ensemble methods are applied for the FEM-based forecast and analysis of the process state. Test cases have shown that, at least for the considered scenarios, only a small number of ensemble members is needed to get a sufficiently good state estimate. Thereby, it has been shown that the computation time compared to the use of the Kalman Filter could be drastically reduced for moderate and large problem dimensions. It has been further stated that the application of a state-dependent model error covariance matrix in the considered applications could be suitable. A method to efficiently handle such a matrix in an ensemble setting has been proposed and, depending on the scenario, results obtained with this variant provide slightly better results than the ones obtained with the popular multiplicative inflation method. For vehicle control, an existing cooperative optimization-based feedback controller has been applied. Requiring specific target points as input, a sequential procedure identifying interesting measurement locations based on the current estimates and their error variances is proposed. To reduce uncertainty on the way to the target points, local appealing points are introduced pulling sensor vehicles to neighboring regions corresponding to higher uncertainty. While this does not lead to an additional significant error reduction in all scenarios, the estimation quality of scenarios on larger spatial scales or with specific shapes of the contaminant cloud highly profits from their use. The proposed monitoring approaches further allow the estimation of the

location of the pollution source together with the dispersion process state. Even if the source location is not known exactly, a suitable choice of representative radial basis functions in combination with joint state-parameter estimation leads to a good approximation of the actual source function. However, it has to be stated that joint state-parameter estimation in the considered context highly depends on the choice of estimation error parameters and is prone to fluctuations. In general, test cases have shown that the estimate obtained with the centralized monitoring approach are much better than the ones obtained with other basic approaches. Its drawback is the computational effort, which increases largely with an increasing number of sensor vehicles and a larger problem dimension. Moreover, the approach requires a robust communication system with high range and bandwidth as well as the availability of a powerful central supercomputer.

In **Chapter 3**, a new decentralized monitoring approach has been proposed. Avoiding the direct integration of a central supercomputer into the system, sensor vehicles are assumed to be equipped with a computation unit and process local information exchanging information with neighbors residing within their communication range. To account for the limited onboard computational capability and to further pursue aspects related to efficiency, Proper Orthogonal Decomposition is applied in this context to generate a reduced order model. A specific choice of the required snapshot set is proposed consisting of simulation results for different initial conditions and impulse response snapshots corresponding to an excitation at the source location. Furthermore, impulse response snapshots corresponding to eigenvectors of a potential model error covariance matrix can be added to account for the preservation of representativeness jeopardized by model error. Test cases have shown that the latter choice is especially beneficial if strong perturbations are considered. However, they are counterproductive in scenarios with low perturbation. Due to different estimates at different nodes of the sensor network, conventional data assimilation methods cannot be used in the considered scenario. Instead, a two-step method operating on the reduced models and consisting of Kalman Filter and Covariance Intersection is applied. The sequential target point generation procedure is modified for the decentralized scenario and an existing variant of the vehicle controller is employed to finalize the monitoring approach. The estimation results for the dispersion process obtained with the decentralized approach are not as good as its centralized counterpart, predominantly caused by vehicles not coordinated on a global level, a longer time to spread information in the network, and the use of reduced models and conservative data assimilation methods. However, the deviation between the results obtained with the centralized approach and with the decentralized approach is not large and usually decreases with proceeding application time. Thus, if a central supercomputer or a communication system with the high requirements of the centralized approach is not available and more emphasis is put on scalability and robustness, the second approach is beneficial. Slight losses in the accuracy of the estimates have to be tolerated. Furthermore, simulation runs to build up the reduced order model have to be performed off-line before the actual application.

A partitioned decentralized monitoring approach has been presented in **Chapter 4**. Especially designed for larger spatial scales and a larger number of sensor vehicles, this approach decomposes the global domain into several subdomains with sensor vehicles being assigned to subdomains they are located in. The idea is that the distributed computing units only have to maintain the local model of the subdomain they are currently located in while communicating with vehicles on neighboring subdomains enables a global convergence of the method. For model prediction and update, an efficient algorithm has been proposed consisting of the application of damped Adaptive Dirichlet-Neumann for the forecast step and a partitioned variant of the Ensemble Square Root Filter for the analysis requiring only little exchange of information between vehicles on neighboring subdomains. While the update part does not introduce additional error to the system, the simplified forecast step does. However, test cases showed that the

error made with the partitioned simulation method can be accepted for the considered Péclet numbers and that they are partially compensated by the integration of new measurements. As the considered processes are dynamic, it should be possible for the sensor vehicles to move from one subdomain to another. Therefore, a modification of the target point generation method and the vehicle controller has been developed allowing the sensor vehicles to move to its neighboring domain if a higher error variance is expected there. However, flexibility of the partitioned approach is limited as the sensor vehicles cannot operate directly globally. Therefore, the quality of the estimates is worse than the quality of the estimates obtained with the centralized approach. Nevertheless, the partitioned approach provides an adequate option for problems with large spatial scale resulting in a distribution of work load. Furthermore, it offers the possibility to set a maximum number of vehicles to be allowed in a domain so that vehicle control can also be considered to be more scalable.

With these new approaches, an alternative to existing monitoring approaches, either based on complex optimal control problems or on heuristic approaches without physical process model, is presented. A middle course between the demand for high-quality estimates and a reduced computational time is steered trying to combine the benefits of both sides. Decentrality offers new possibilities regarding scalability and communication requirements.

This thesis has not put its focus on a single specific application scenario. Instead, a broad class of problems has been considered and general methods have been developed. To apply the approaches of this work in more specific scenarios, adaptions have to be made at some places. For this reason, potential extensibility has been respected during the set up of the monitoring approaches so that it is possible to extend the present approach to more complex or more specialized scenarios. In this way, current limitations like linearity of the PDE models, two-dimensional scenarios and Gaussian errors can be overcome.

While most of the proposed building blocks also work with nonlinear models, besides the choice of the snapshot set, increased computational time would be probably the highest hurdle for the application of nonlinear models.

Increased computation time and memory requirements can be also expected when considering three-dimensional scenarios. However, the three-dimensional extension of the presented monitoring approaches is straightforward: The problem domain has to be discretized by tetrahedrons (or other suitable polyhedrons) and the advection-diffusion solver has to be adapted accordingly. While the resulting state vector possesses a highly increased dimension making the application of reduced order models in this context even more beneficial, it can be handled in the further monitoring steps in the same way as its two-dimensional counterpart.

The approaches have been evaluated on rectangular problem domains. However, the developed approaches are not limited to such simple domains and can be applied on much more complex domains. The FEM-based advection-diffusion solver can naturally deal with more complex domains and the vehicle controller would maintain a number of additional constraints to prevent the vehicles from leaving the considered domain.

Other error distributions would need other data assimilation methods. A possible adaption would be to consider the error to be log-normally distributed. In this way, the current estimation procedure could be mostly preserved since the logarithm of the random variable is again Gaussian.

Besides, further extensions are possible in future. Currently, the velocity field is assumed to be known. However, if the velocity field is unknown, one could estimate the velocity field along with the concentration – either by parameterizing the velocity and treating it with an additional parameter set or with

sensors also measuring velocity and estimating the global velocity at every grid point. In a similar way, diffusion coefficients could be estimated.

It is conceivable that reaction terms could be also included in future work when dealing with dispersion processes highly affected by such kind of processes. An extension of the advection-diffusion equation would be necessary and an advection-diffusion-reaction equation would have to be solved.

The developed approaches can not only be applied in atmospheric dispersion monitoring, but in a number of other scenarios, where advection-diffusion processes are dominant. This includes monitoring of submarine processes, e.g. after oil spills, numerical weather prediction and monitoring of wildland fires. In these scenarios, additional measurements of the process state could be gathered at important locations contributing to an improved weather forecast or disaster assessment.

The structure of the presented monitoring approach generally allows the exchange of individual building blocks. For example, the approach could be tested with other data assimilation strategies, other methods for identifying target points, another vehicle controller, etc. In particular, it would be very interesting and beneficial concerning computation time if a reduced order model is applied together with the partitioned approach.

The focus of the present work has been set on the development of new efficient approaches, algorithms and methods, less on the actual optimization of the resulting code. Thus, further speed-up by code optimization is possible and, consequently, even more efficient monitoring approaches can be obtained. This opens the door for the implementation of the proposed approaches on practical sensor vehicles, and, thereby, to the possibility to apply and validate the presented monitoring approaches in practical experiments.

List of Acronyms

ADN	Adaptive Dirichlet-Neumann
AO	Adaptive Observation
BT	Balanced Truncation
CI	Covariance Intersection
CPS	Cyber-Physical System
d-ADN	damped Adaptive Dirichlet Neumann
DPS	Distributed Parameter System
DDDAS	Dynamic Data-Driven Application System
EnKF	Ensemble Kalman Filter
EnSRF	Ensemble Square Root Filter
FEM	Finite Element Method
FIM	Fisher Information Matrix
KF	Kalman Filter
LTI	Linear Time-Invariant
MILP	Mixed Integer Linear Program
MLD	Mixed Logical Dynamical
MPC	Model Predictive Control
NWP	Numerical Weather Prediction
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
POD	Proper Orthogonal Decomposition
RBF	Radial Basis Function
ROM	Reduced Order Model
UAV	Unmanned Aerial Vehicle

List of Symbols (Selection)

a	Scalar or scalar function
A	Scalar or set
$\mathbf{a}$	Vector or vector-valued function
$\mathbf{A}$	Matrix
$\mathbf{a}_i$	i-th variant, part, etc. of $\mathbf{a}$
$\mathbf{a_i}$	Individual entity $\mathbf{a_i}$
$\mathbf{a}^{(i)}$	$\mathbf{a}$ at time $t = i \cdot \Delta t$
$\mathbf{a}_{(i)}$	i-th ensemble member variant of $\mathbf{a}$
$(\mathbf{a})_i$	i-th component of $\mathbf{a}$
$\mathbf{a^a}$	Analysis of $\mathbf{a}$
$\mathbf{a^c}$	Augmented with entities $\mathbf{a}_i$ of neighboring subdomains Ω_i
$\mathbf{a^f}$	Forecast of $\mathbf{a}$
$\mathbf{a^t}$	True $\mathbf{a}$
$\mathbf{a}^-$	$\mathbf{a}$ after first analysis step
$\mathbf{a}^+$	$\mathbf{a}$ after second analysis step
$\hat{\mathbf{a}}$	Reduced variant of $\mathbf{a}$
$\tilde{\mathbf{a}}$	(Augmented) variant of $\mathbf{a}$
$\bar{\mathbf{a}}$	Mean of $\mathbf{a}$
$\mathbf{A}^T$	Transpose of $\mathbf{A}$
$\mathbf{A}^{-1}$	Inverse of $\mathbf{A}$
$\mathbb{1}$	Identity matrix
∇	Nabla operator
Δ	Laplace operator
$\|\mathbf{a}\|$	Euclidean norm of $\mathbf{a}$
$\mathbf{a} \cdot \mathbf{b}$	Scalar product (if declared between two vectors)
$\mathbf{A} \circ \mathbf{B}$	Schur product
$\emptyset$	Empty set
$\text{col}\{\mathbf{a}_1, \mathbf{a}_2\}$	$\begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{pmatrix}$
$\text{row}\{\mathbf{a}_1, \mathbf{a}_2\}$	$\begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_2 \end{pmatrix}$
$\text{blockdiag}\{\mathbf{A}_1, \mathbf{A}_2\}$	$\begin{pmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 \end{pmatrix}$

n	Dimension of state vector $\mathbf{x}$, number of grid points
n_θ	Dimension of parameter vector $\boldsymbol{\theta}$, number of parameters
n_T	Number of time steps
n_p	Number of observations
n_s	Number of ensemble members
n_ℓ	Dimension of the reduced state vector $\hat{\mathbf{x}}$, number of POD bases
n_S	Number of snapshot simulations
n_C	Number of snapshot elements
n_{SS}	Number of representative source functions
n_{IC}	Number of representative initial conditions
n_η	Number of eigenvectors of $\mathbf{Q}$
n_d	Number of subdomains
$n_{\Gamma_{ij}}$	Number of grid points at boundary Γ_{ij}
n_V	Number of vehicles
n_M	Number of target points
n_γ	Number of hyperplanes to approximate distance functions
n_{Tr}	Number of time steps for reduced MPC time horizon
$\mathbf{r} = (r_1, r_2)^T$	Spatial vector
x, $\mathbf{x}$	State, state vector; associated with pollutant concentration
t	Time
$\mathbf{w} = (w_1, w_2)^T$	Process velocity
D, $\mathbf{D}$	Diffusivity, diffusion matrix
s	Source function
Ω	Domain
$\partial\Omega$	Boundary of domain Ω
Γ_{ij}	Interface between subdomain Ω_i and Ω_j
$\mathbf{n}$	Normal vector
Δr	Grid spacing
Δt	Time step size
ϕ, $\boldsymbol{\phi}$	Basis function, basis vector
$\mathbf{r_c}$	Characteristic function
θ, $\boldsymbol{\theta}$	Parameter, parameter vector; associated with source output
q	Spatial source function
$\mathbf{S}$	Discretized spatial source function
$\mathbf{C}$	Snapshot matrix
δ	Fraction of snapshot set energy covered with reduced basis
$\mathbf{\Phi}$	POD matrix
$\mathbf{y}$	Observation
$\mathbf{H}$	Observation model matrix

$\boldsymbol{\epsilon}$	Observation error
$\mathbf{R}$	Observation error covariance matrix
$\mathbf{M}$	Forecast model matrix
$\boldsymbol{\eta}$	Model error
$\mathbf{Q}$	Model error covariance matrix
$\mathbf{P}$	Estimate error covariance matrix
$\mathbf{K}$	Kalman gain
$\mathbf{d}$	Observation innovation
$\mathbf{X}$	Ensemble deviation matrix
$\mathbf{Z}$	Ensemble deviation matrix at observation site
$\boldsymbol{\Psi_1}$, $\boldsymbol{\Psi_2}$	Localization matrices
α	Inflation factor
$\boldsymbol{\Upsilon}$	Square root of state-dependent model error covariance matrix
$\mathbf{i}$	Information vector
$\mathbf{I}$	Information matrix
$\mathbf{p}$, $\mathbf{p_g}$	Vehicle state, global vehicle state
$\mathbf{r^p}$	Vehicle position
$\mathbf{v^p}$	Vehicle velocity
$\mathbf{u}$, $\mathbf{u_g}$	Control input, global control input
U_i	Set of input control vectors for i times
$\mathbf{q}$, $\mathbf{q_g}$	Target point state, global target point state
d	Communication range
N_i	Set of sensors within communication range d of vehicle i
$N_i^\Omega = N_j^{\Gamma_j i}$	Set of sensors on subdomain Ω_i
$\mathbf{x}^{\Gamma_{ij}}$	State vector at boundary Γ_{ij}

List of Figures

Bibliography

[1] V. Akçelik, G. Biros, A. Draganescu, J. Hill, O. Ghattas, and B. V. B. Waanders. Dynamic data-driven inversion for terascale simulations: Real-time identification of airborne contaminants. In *Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*, page 43. IEEE Computer Society, 2005.

[2] A. Aksoy, F. Zhang, and J. W. Nielsen-Gammon. Ensemble-based simultaneous state and parameter estimation in a two-dimensional sea-breeze model. *Monthly Weather Review*, 134(10):2951–2970, 2006.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[4] J. Allred, A. B. Hasan, S. Panichsakul, W. Pisano, P. Gray, J. Huang, R. Han, D. Lawrence, and K. Mohseni. SensorFlock: an airborne wireless sensor network of micro-air vehicles. In *Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 117–129. ACM, 2007.

[5] A. A. Alonso, I. G. Kevrekidis, J. R. Banga, and C. E. Frouzakis. Optimal sensor location and reduced order observer design for distributed process systems. *Computers & chemical engineering*, 28(1):27–35, 2004.

[6] M. E. Alpay and M. H. Shor. Model-based solution techniques for the source localization problem. *IEEE Transactions on Control Systems Technology*, 8:895–904, 2000.

[7] J. L. Anderson. An ensemble adjustment Kalman filter for data assimilation. *Monthly Weather Review*, 129(12):2884–2903, 2001.

[8] J. L. Anderson and S. L. Anderson. A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, 127(12):2741–2758, 1999.

[9] J. Annan, J. Hargreaves, N. Edwards, and R. Marsh. Parameter estimation in an intermediate complexity earth system model using an ensemble Kalman filter. *Ocean Modelling*, 8(1):135–154, 2005.

[10] A. C. Antoulas. *Approximation of large-scale dynamical systems*, volume 6. Siam, 2005.

[11] S. P. Arya. *Air pollution meteorology and dispersion*. Oxford University Press New York, 1999.

[12] J. Baiges, R. Codina, and S. Idelsohn. A domain decomposition strategy for reduced order models. application to the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 267:23–42, 2013.

[13] G. Battistelli, L. Chisci, N. Forti, G. Pelosi, and S. Selleri. Distributed finite element Kalman filter. In *Proceedings of the European Control Conference (ECC)*, pages 3695–3700. IEEE, 2015.

[14] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35:407–427, 1999.

[15] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4), 2015.

[16] V. H. Bennetts, A. J. Lilienthal, P. P. Neumann, and M. Trincavelli. Mobile robots for localizing gas emission sources on landfill sites: is bio-inspiration the way to go? *Frontiers in Neuroengineering*, 4:20, 2012.

[17] M. R. Beychok. *Fundamentals of Stack Gas Dispersion: Guide*. 1994.

[18] C. H. Bishop. The GIGG-EnKF: ensemble Kalman filtering for highly skewed non-negative uncertainty distributions. *Quarterly Journal of the Royal Meteorological Society*, 142(696):1395–1412, 2016.

[19] C. H. Bishop, B. J. Etherton, and S. J. Majumdar. Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. *Monthly Weather Review*, 129(3):420–436, 2001.

[20] C. H. Bosanquet and J. L. Pearson. The spread of smoke and gases from chimneys. *Transactions of the Faraday Society*, 32(0):1249–1263, 1936.

[21] F. Bouttier and P. Courtier. Data assimilation concepts and methods march 1999. *Meteorological Training Course Lecture Series. ECMWF*, 2002.

[22] J. Brink and E. Pebesma. Plume tracking with a mobile sensor based on incomplete and imprecise information. *Transactions in GIS*, 18(5):740–766, 2014.

[23] A. N. Brooks and T. J. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1-3):199–259, 1982.

[24] R. Buizza and A. Montani. Targeting observations using singular vectors. *Journal of the Atmospheric Sciences*, 56(17):2965–2985, 1999.

[25] R. Buizza and T. N. Palmer. The singular-vector structure of the atmospheric global circulation. *Journal of the Atmospheric Sciences*, 52(9):1434–1456, 1995.

[26] F. Bullo, J. Cortés, and S. Martínez. *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. Princeton University Press, 2009.

[27] G. Burgers, P. Jan van Leeuwen, and G. Evensen. Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*, 126(6):1719–1724, 1998.

[28] J. Burns, E. Cliff, and C. Rautenberg. A distributed parameter control approach to optimal filtering and smoothing with mobile sensor networks. In *Proceedings of the 17th Mediterranean Conference on Control and Automation*, pages 181–186. IEEE, 2009.

[29] M. E. Campbell and N. R. Ahmed. Distributed data fusion: Neighbors, rumors, and the art of collective knowledge. *IEEE Control Systems*, 36(4):83–109, 2016.

[30] C. Carlenzoli and A. Quarteroni. Adaptive domain decomposition methods for advection-diffusion problems. In *Modeling, mesh generation, and adaptive numerical methods for partial differential equations*, pages 165–186. Springer, 1995.

[31] L. Carotenuto, P. Muraca, and G. Raiconi. Optimal location of a moving sensor for the estimation of a distributed-parameter process. *International Journal of Control*, 46(5):1671–1688, 1987.

[32] F. S. Cattivelli and A. H. Sayed. Diffusion strategies for distributed Kalman filtering and smoothing. *IEEE Transactions on Automatic Control*, 55(9):2069–2084, 2010.

[33] Z. Chen. *Finite element methods and their applications*. Springer, Berlin; New York, 2005.

[34] H.-L. Choi and J. P. How. Continuous trajectory planning of mobile sensors for informative forecasting. *Automatica*, 46(8):1266–1275, 2010.

[35] V. N. Christopoulos and S. Roumeliotis. Multi robot trajectory generation for single source explosion parameter estimation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2803–2809. IEEE, 2005.

[36] A. J. Cimorelli, S. G. Perry, A. Venkatram, J. C. Weil, R. J. Paine, R. B. Wilson, R. F. Lee, W. D. Peters, and R. W. Brode. AERMOD: A dispersion model for industrial source applications. Part I: General model formulation and boundary layer characterization. *Journal of Applied Meteorology*, 44(5):682–693, 2005.

[37] C. Corrigan, G. Roberts, M. Ramana, D. Kim, and V. Ramanathan. Capturing vertical profiles of aerosols and black carbon over the indian ocean using autonomous unmanned aerial vehicles. *Atmospheric Chemistry and Physics Discussions*, 7(4):11429–11463, 2007.

[38] X. Cui, T. Hardin, R. K. Ragade, and A. S. Elmaghraby. A swarm-based fuzzy logic control mobile sensor network for hazardous contaminants localization. In *Proceedings of the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004)*, 2004.

[39] D. N. Daescu and I. M. Navon. Adaptive observations in the context of 4D-Var data assimilation. *Meteorology and Atmospheric Physics*, 85(4):205–226, 2003.

[40] K. Daniel, B. Dusza, A. Lewandowski, and C. Wietfeld. AirShield: A system-of-systems MUAV remote sensing architecture for disaster response. In *Proceedings of the 3rd Annual IEEE Systems Conference*, pages 196–200. IEEE, 2009.

[41] F. Darema. Dynamic data driven applications systems: A new paradigm for application simulations and measurements. In *Proceedings of the International Conference on Computational Science (ICCS)*, pages 662–669. Springer, 2004.

[42] F. Darema. Grid computing and beyond: The context of dynamic data driven applications systems. *Proceedings of the IEEE*, 93(3):692–697, 2005.

[43] M. Demetriou. Power management of sensor networks for detection of a moving source in 2-d spatial domains. In *Proceedings of the American Control Conference (ACC)*, page 6 pp., 2006.

[44] M. Demetriou. Guidance of mobile actuator-plus-sensor networks for improved control and estimation of distributed parameter systems. *IEEE Transactions on Automatic Control*, 55(7):1570–1584, 2010.

[45] M. Demetriou and D. Uciński. State estimation of spatially distributed processes using mobile sensing agents. In *Proceedings of the American Control Conference (ACC)*, pages 1770–1776, 2011.

[46] M. A. Demetriou, N. A. Gatsonis, and J. R. Court. Coupled controls-computational fluids approach for the estimation of the concentration from a moving gaseous source in a 2-d domain with a Lyapunov-guided sensing aerial vehicle. *IEEE Transactions on Control Systems Technology*, 22(3):853–867, 2014.

[47] Z. Deng, P. Zhang, W. Qi, J. Liu, and Y. Gao. Sequential covariance intersection fusion Kalman filter. *Information Sciences*, 189:293–309, 2012.

[48] F.-X. L. Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A*, 38A(2):97–110, 1986.

[49] J. Donea. A Taylor-Galerkin method for convective transport problems. *International Journal for Numerical Methods in Engineering*, 20(1):101–119, 1984.

[50] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12(656–704):3, 2009.

[51] R. Draxler and G. Hess. *Description of the HYSPLIT-4 Modeling System*. NOAA technical memorandum ERL ARL. National Oceanic and Atmospheric Administration, 1997.

[52] M. Duckham, S. Nittel, and M. Worboys. Monitoring dynamic spatial fields using responsive geosensor networks. In *Proceedings of the 13th Annual ACM International Workshop on Geographic information systems*, GIS '05, pages 51–60. ACM, 2005.

[53] M. Dunbabin and L. Marques. Robotics for environmental monitoring: Significant advancements and applications. *IEEE Robotics & Automation Magazine*, 19(1):24–39, 2012.

[54] J. Euler. *Optimal Cooperative Control of UAVs for Dynamic Data-Driven Monitoring Tasks*. PhD thesis, Technische Universität Darmstadt, Department of Computer Science, 2017.

[55] J. Euler, A. Horn, D. Haumann, J. Adamy, and O. von Stryk. Cooperative n-boundary tracking in large scale environments. In *Proceedings of the 6th International Workshop on Wireless Sensor, Actuator and Robot Networks*, volume 7, pages 425–443, 2012.

[56] J. Euler, T. Ritter, S. Ulbrich, and O. von Stryk. Centralized ensemble-based trajectory planning of cooperating sensors for estimating atmospheric dispersion processes. In *Dynamic Data-Driven Environmental Systems Science*, volume 8964 of *Lecture Notes in Computer Science*, pages 322–333. Springer International Publishing, 2015.

[57] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99(C5):10143–10162, 1994.

[58] G. Evensen. The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, 53(4):343–367, 2003.

[59] G. Evensen. Sampling strategies and square root analysis schemes for the EnKF. *Ocean Dynamics*, 54(6):539–560, 2004.

[60] G. Evensen, D. P. Dee, and J. Schröter. Parameter estimation in dynamical models. In *Ocean Modeling and Parameterization*, pages 373–398. Springer, 1998.

[61] M. Farina, G. Ferrari-Trecate, and R. Scattolini. Moving-horizon partition-based state estimation of large-scale systems. *Automatica*, 46(5):910–918, 2010.

[62] E. Fiorelli, N. E. Leonard, S. Member, P. Bhatta, D. A. Paley, S. Member, R. Bachmayer, and D. M. Fratantoni. Multi-AUV control and adaptive sampling in monterey bay. In *IEEE Journal of Oceanic Engineering*, pages 935–948, 2004.

[63] G. Gaspari and S. E. Cohn. Construction of correlation functions in two and three dimensions. *Quarterly Journal of the Royal Meteorological Society*, 125(554):723–757, 1999.

[64] F. Gastaldi, L. Gastaldi, and A. Quarteroni. Adaptive domain decomposition methods for advection dominated equations. *East West Journal of Numerical Mathematics*, 4:165–206, 1996.

[65] N. A. Gatsonis, M. A. Demetriou, and T. Egorova. Real-time prediction of gas contaminant concentration from a ground intruder using a UAV. In *Proceedings of the IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–6. IEEE, 2015.

[66] A. Gelb. *Applied optimal estimation*. MIT press, 1974.

[67] J. Ghiglieri and S. Ulbrich. Optimal flow control based on pod and mpc and an application to the cancellation of Tollmien-Schlichting waves. *Optimization Methods and Software*, 29(5):1042–1074, 2014.

[68] S. Gillijns, O. Mendoza, J. Chandrasekar, B. L. R. De Moor, D. Bernstein, and A. Ridley. What is the ensemble Kalman filter and how well does it work? In *Proceedings of the American Control Conference (ACC)*, page 6 pp., 2006.

[69] V. Girault and P.-A. Raviart. *Finite element approximation of the Navier-Stokes equations*, volume 749. Springer (Lecture Notes in Mathematics), 1979.

[70] N. J. Gordon, D. J. Salmond, and A. F. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F-Radar and Signal Processing*, 140(2):107–113, 1993.

[71] B. Grocholsky. Information-theoretic control of multiple sensor platforms, 2002.

[72] M. T. Gudmundsson, T. Thordarson, Á. Höskuldsson, G. Larsen, H. Björnsson, F. J. Prata, B. Oddsson, E. Magnússon, T. Högnadóttir, G. N. Petersen, et al. Ash generation and distribution from the april-may 2010 eruption of eyjafjallajökull, iceland. *Scientific Reports*, 2:572, 2012.

[73] T. M. Hamill, J. S. Whitaker, and C. Snyder. Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter. *Monthly Weather Review*, 129(11):2776–2790, 2001.

[74] S. R. Hanna, G. A. Briggs, and R. P. Hosker. *Handbook on atmospheric diffusion*. Technical Information Center, U.S. Dept. of Energy, [Oak Ridge, TN], 1982.

[75] J. Haugen and L. Imsland. Monitoring an advection-diffusion process using aerial mobile sensors. *Unmanned Systems*, 3(03):221–238, 2015.

[76] A. T. Hayes, A. Martinoli, and R. M. Goodman. Distributed odor source localization. *IEEE Sensors Journal*, 2(3):260–271, 2002.

[77] M. Hinze and S. Volkwein. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control. In *Dimension Reduction of Large-Scale Systems*, pages 261–306. Springer, 2005.

[78] N. Holmes and L. Morawska. A review of dispersion modelling and its application to the dispersion of particles: An overview of different dispersion models available. *Atmospheric Environment*, 40(30):5902–5928, 2006.

[79] P. Houtekamer and F. Zhang. Review of the ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 144(12):4489–4532, 2016.

[80] P. L. Houtekamer and H. L. Mitchell. Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review*, 126(3):796–811, 1998.

[81] P. L. Houtekamer and H. L. Mitchell. A sequential ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 129(1):123–137, 2001.

[82] F. S. Hover. Path planning for data assimilation in mobile environmental monitoring systems. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IROS'09, pages 213–218. IEEE Press, 2009.

[83] B. R. Hunt, E. J. Kostelich, and I. Szunyogh. Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D: Nonlinear Phenomena*, 230(1-2):112–126, 2007.

[84] P. Hurley. PARTPUFF – a Lagrangian particle-puff approach for plume dispersion modeling applications. *Journal of Applied Meteorology*, 33(2):285–294, 1994.

[85] P. Hurley. TAPM V4. Part 1: Technical description, 2008.

[86] IBM ILOG CPLEX V12.1. User's manual for CPLEX, 2009.

[87] H. Ishida, G. Nakayama, T. Nakamoto, and T. Moriizumi. Controlling a gas/odor plume-tracking robot based on transient responses of gas sensors. *IEEE Sensors Journal*, 5(3):537–545, 2005.

[88] A. H. Jazwinski. *Stochastic processes and filtering theory*. Courier Corporation, 2007.

[89] Z. Jin and A. L. Bertozzi. Environmental boundary tracking and estimation using multiple autonomous vehicles. In *Proceedings of the 46th IEEE Conference on Decision and Control (CDC)*, pages 4918–4923. IEEE, 2007.

[90] B. J. Julian, M. Angermann, M. Schwager, and D. Rus. Distributed robotic sensor networks: An information-theoretic approach. *International Journal of Robotics Research*, 31(10):1134–1154, 2012.

[91] S. Julier and J. Uhlmann. A non-divergent estimation algorithm in the presence of unknown correlations. In *Proceedings of the American Control Conference (ACC)*, volume 4, pages 2369–2373, 1997.

[92] S. J. Julier and J. K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Proceedings of Aerosense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, pages 182–193, 1997.

[93] S. J. Julier and J. K. Uhlmann. General decentralized data fusion with covariance intersection. *Handbook of Multisensor Data Fusion*, pages 319–342, 2001.

[94] M. Kaazempur-Mofrad and C. Ethier. An efficient characteristic Galerkin scheme for the advection equation in 3-d. *Computer Methods in Applied Mechanics and Engineering*, 191(46):5345–5363, 2002.

[95] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.

[96] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Journal of Fluids Engineering*, 83(1):95–108, 1961.

[97] H. Karimipour and V. Dinavahi. Parallel domain decomposition based distributed state estimation for large-scale power systems. In *Proceedings of the 51st IEEE/IAS Industrial & Commercial Power Systems Technical Conference (I&CPS)*, pages 1–5. IEEE, 2015.

[98] C. L. Keppenne. Data assimilation into a primitive-equation model with a parallel ensemble Kalman filter. *Monthly Weather Review*, 128(6):1971–1981, 2000.

[99] A. Khamis, A. Hussein, and A. Elmogy. Multi-robot task allocation: A review of the state-of-the-art. In *Cooperative Robots and Sensor Networks 2015*, pages 31–51. Springer, 2015.

[100] U. A. Khan and J. M. Moura. Distributing the Kalman filter for large-scale systems. *IEEE Transactions on Signal Processing*, 56(10):4919–4935, 2008.

[101] J. Kho, A. Rogers, and N. R. Jennings. Decentralized control of adaptive sampling in wireless sensor networks. *ACM Transactions on Sensor Networks*, 5(3):19:1–19:35, 2009.

[102] C. S. Kubrusly and H. Malebranche. Sensors and controllers location in distributed systems-a survey. *Automatica*, 21(2):117–128, 1985.

[103] J. Kuhn. Model-predictive control of cooperative multi-vehicle systems based on discrete-time linear systems. Master's thesis, Technische Universität Darmstadt, Department of Computer Science & Department of Mathematics, 2009.

[104] J. Kuhn, C. Reinl, and O. von Stryk. Predictive control for multi-robot observation of multiple moving targets based on discrete-continuous linear models. In *Proceedings of the 18th IFAC World Congress*, pages 257–262, 2011.

[105] Y. Kuroki, G. S. Young, and S. E. Haupt. UAV navigation by an expert system for contaminant mapping with a genetic algorithm. *Expert Systems with Applications*, 37(6):4687–4697, 2010.

[106] Y. Kuwana, I. Shimoyama, and H. Miura. Steering control of a mobile robot using insect antennae. In *IEEE International Conference on Intelligent Robots and Systems*, volume 2, pages 530–535. IEEE, 1995.

[107] R. H. Langland and G. D. Rohaly. Adjoint-based targeting of observations for fastex cyclones. Technical report, DTIC Document, 1996.

[108] E. A. Lee. Cyber physical systems: Design challenges. In *Proceedings of 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pages 363–369, 2008.

[109] N. Leonard, D. Paley, F. Lekien, R. Sepulchre, D. Fratantoni, and R. Davis. Collective motion, sensor networks, and ocean sampling. *Proceedings of the IEEE*, 95(1):48–74, 2007.

[110] J. M. Lewis, S. Lakshmivarahan, and S. Dhall. *Dynamic Data Assimilation: A Least Squares Approach*. Cambridge University Press, 2006.

[111] W. Li and C. G. Cassandras. A cooperative receding horizon controller for multivehicle uncertain environments. *IEEE Transactions on Automatic Control*, 51(2):242–257, 2006.

[112] A. Lilienthal, D. Reimann, and A. Zell. Gas source tracing with a mobile robot using an adapted moth strategy. In *Autonome Mobile Systeme 2003*, pages 150–160. Springer, 2003.

[113] A. C. Lorenc. Analysis methods for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 112(474):1177–1194, 1986.

[114] E. N. Lorenz and K. A. Emanuel. Optimal sites for supplementary weather observations: Simulation with a small model. *Journal of Atmospheric Sciences*, 55:399–414, 1998.

[115] J. L. Lumley. The Structure of Inhomogeneous Turbulent Flows. In A. M. Yaglom and V. I. Tatarski, editors, *Atmospheric Turbulence and Radio Propagation*, pages 166–178. Nauka, 1967.

[116] R. Madankan, P. Singla, and T. Singh. Optimal information collection for source parameter estimation of atmospheric release phenomenon. In *Proceedings of the American Control Conference (ACC)*, pages 604–609. IEEE, 2014.

[117] S. J. Majumdar, C. H. Bishop, B. J. Etherton, and Z. Toth. Adaptive sampling with the ensemble transform Kalman filter. Part II: field program implementation. *Monthly Weather Review*, 130(5):1356–1369, 2002.

[118] J. Mandel, L. S. Bennethum, J. D. Beezley, J. L. Coen, C. C. Douglas, M. Kim, and A. Vodacek. A wildland fire model with data assimilation. *Mathematics and Computers in Simulation*, 79(3):584–606, 2008.

[119] L. Marques, U. Nunes, and A. T. de Almeida. Olfaction-based mobile robot navigation. *Thin Solid Films*, 418(1):51–58, 2002.

[120] S. Martínez, J. Cortés, and F. Bullo. Motion coordination with distributed information. *IEEE Control Systems*, 27(4):75–88, 2007.

[121] R. McCune, R. Purta, M. Dobski, A. Jaworski, G. Madey, Y. Wei, A. Madey, and M. B. Blake. Investigations of dddas for command and control of UAV swarms with agent-based modeling.

In *Proceedings of the 2013 Winter Simulation Conference: Simulation: Making Decisions in a Complex World*, pages 1467–1478. IEEE Press, 2013.

[122] A. Milionis and T. Davies. Regression and stochastic models for air pollution – I. review, comments and suggestions. *Atmospheric Environment*, 28(17):2801–2810, 1994.

[123] D. Moon, A. Albergel, F. Jasmin, and G. Thibaut. The use of the MERCURE CFD code to deal with an air pollution problem due to building wake effects. *Journal of Wind Engineering and Industrial Aerodynamics*, 67-68:781–791, 1997.

[124] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE transactions on automatic control*, 26(1):17–32, 1981.

[125] H. Moradkhani, S. Sorooshian, H. V. Gupta, and P. R. Houser. Dual state-parameter estimation of hydrological models using ensemble Kalman filter. *Advances in Water Resources*, 28(2):135–147, 2005.

[126] A. G. Mutambara. *Decentralized estimation and control for multisensor systems*. CRC press, 1998.

[127] R. Negenborn and J. Maestre. Distributed model predictive control: An overview and roadmap of future research opportunities. *IEEE Control Systems*, 34(4):87–97, 2014.

[128] A. Nehorai, B. Porat, and E. Paldi. Detection and localization of vapor-emitting sources. *IEEE Transactions on Signal Processing*, 43(1):243–253, 1995.

[129] B. Noack, J. Sijs, and U. D. Hanebeck. Fusion strategies for unequal state vectors in distributed Kalman filtering. *IFAC Proceedings Volumes*, 47(3):3262–3267, 2014.

[130] OECD. *Emerging Risks in the 21st Century An Agenda for Action: An Agenda for Action*. OECD Publishing, 2003.

[131] R. Olfati-Saber. Distributed Kalman filter with embedded consensus filters. In *Proceedings of the 44th IEEE Conference on Decision and Control (CDC)*, pages 8179–8184, 2005.

[132] E. Ott, B. R. Hunt, I. Szunyogh, A. V. Zimin, E. J. Kostelich, M. Corazza, E. Kalnay, D. Patil, and J. A. Yorke. A local ensemble Kalman filter for atmospheric data assimilation. *Tellus A*, 56(5):415–428, 2004.

[133] T. N. Palmer, R. Gelaro, J. Barkmeijer, and R. Buizza. Singular vectors, metrics, and adaptive observations. *Journal of the Atmospheric Sciences*, 55(4):633–653, 1998.

[134] L. Peng, D. Lipinski, and K. Mohseni. Dynamic data driven application system for plume estimation using UAVs. *Journal of Intelligent & Robotic Systems*, 74(1-2):421–436, 2014.

[135] L. Peng, M. Silic, and K. Mohseni. A dddas plume monitoring system with reduced Kalman filter. In *Proceedings of the International Conference on Computational Science (ICCS)*, volume 51, pages 2533–2542. Elsevier, 2015.

[136] P.-O. Persson and G. Strang. A simple mesh generator in Matlab. *SIAM review*, 46(2):329–345, 2004.

[137] D. T. Pham. Stochastic methods for sequential data assimilation in strongly nonlinear systems. *Monthly Weather Review*, 129(5):1194–1207, 2001.

[138] O. Pironneau. On the transport-diffusion algorithm and its applications to the Navier-Stokes equations. *Numerische Mathematik*, 38(3):309–332, 1982.

[139] C. Prud'homme, D. V. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output

bound methods. *Journal of Fluids Engineering*, 124(1):70–80, 2002.

[140] Z.-X. Pu, E. Kalnay, J. Sela, and I. Szunyogh. Sensitivity of forecast errors to initial conditions with a quasi-inverse linear method. *Monthly Weather Review*, 125(10):2479–2503, 1997.

[141] A. Quarteroni and A. Valli. *Domain decomposition methods for partial differential equations*. Oxford University Press, 1999.

[142] E. Rafajłowicz. Optimum choice of moving sensor trajectories for distributed-parameter system identification. *International Journal of Control*, 43(5):1441–1451, 1986.

[143] E. Ragnoli, S. Zhuk, M. Zayats, and M. Hartnett. Domain decomposition for a linear advection-diffusion equation by means of minimax filtering. In *Proceedings of the European Control Conference (ECC)*, 2014.

[144] E. Ragnoli, S. Zhuk, M. Zayats, and M. Hartnett. Localised filtering for transport models. In *Proceedings of the 54th IEEE Conference on Decision and Control (CDC)*, pages 449–454. IEEE, 2015.

[145] K. S. Rao. Source estimation methods for atmospheric dispersion. *Atmospheric Environment*, 1(33):6964–6973, 2007.

[146] C. Reinl. *Trajektorien- und Aufgabenplanung kooperierender Fahrzeuge: Diskret-kontinuierliche Modellierung und Optimierung (Trajectory and Task Planning for Cooperating Vehicles: Discrete-Continuous Modeling and Optimization)*. PhD thesis, Technische Universität Darmstadt, Department of Computer Science, 2010.

[147] T. Ritter, J. Euler, S. Ulbrich, and O. von Stryk. Adaptive observation strategy for dispersion process estimation using cooperating mobile sensors. In *Proceedings of the 19th IFAC World Congress*, pages 5302 – 5308, 2014.

[148] T. Ritter, J. Euler, S. Ulbrich, and O. von Stryk. Decentralized dynamic data-driven monitoring of atmospheric dispersion processes. *Procedia Computer Science*, 80:919 – 930, 2016.

[149] T. Ritter, S. Ulbrich, and O. von Stryk. Decentralized dynamic data-driven monitoring of dispersion processes on decomposed domains. *Procedia Computer Science*, 108:1632–1641, 2017.

[150] L. A. Rossi, B. Krishnamachari, and C.-C. Kuo. Distributed parameter estimation for monitoring diffusion phenomena using physical models. In *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad-Hoc Communications and Networks*, pages 460–469. IEEE, 2004.

[151] N. Roy, H.-L. Choi, D. Gombos, J. Hansen, J. How, and S. Park. Adaptive observation strategies for forecast error minimization. In *Proceedings of the International Conference on Computational Science (ICCS)*, number 4487 in Lecture Notes in Computer Science, pages 1138–1146. Springer Berlin Heidelberg, 2007.

[152] J. J. Ruiz, M. Pulido, and T. Miyoshi. Estimating model parameters with ensemble-based data assimilation: A review. *Journal of the Meteorological Society of Japan*, 91(2):79–99, 2013.

[153] R. A. Russell, D. Thiel, R. Deveza, and A. Mackay-Sim. A robotic system to locate hazardous chemical leaks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 556–561. IEEE, 1995.

[154] G. Sandini, G. Lucarini, and M. Varoli. Gradient driven self-organizing systems. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, volume 1, pages 429–432. IEEE, 1993.

[155] M. Schäfer. *Computational Engineering – Introduction to Numerical Methods*. Springer Science & Business Media, 2006.

[156] J. S. Scire, D. G. Strimaitis, and R. J. Yamartino. A user's guide for the CALPUFF dispersion model. *Earth Technical Inc., Concord, MA*, 2000.

[157] O. Sigurjonsson. Photos from Eyjafjallajokull. `http://en.vedur.is/earthquakes-and-volcanism/articles/nr/1904`. Accessed: 2017-01-08.

[158] L. Sirovich. Turbulence and the dynamics of coherent structures. Part I: Coherent structures. *Quarterly of Applied Mathematics*, 45(3):561–571, 1987.

[159] I. J. Sledge, L. Peng, and K. Mohseni. An empirical reduced modeling approach for mobile, distributed sensor platform networks. In *Dynamic Data-Driven Environmental Systems Science*, pages 195–204. Springer, 2015.

[160] V. Šmídl and R. Hofman. Tracking of atmospheric release of pollution using unmanned aerial vehicles. *Atmospheric Environment*, 67:425–436, 2013.

[161] P. J. Smith. *Joint state and parameter estimation using data assimilation with application to morphodynamic modelling*. PhD thesis, University of Reading, 2010.

[162] Z. Song, Y. Chen, J. Liang, and D. Uciński. Optimal mobile sensor motion planning under non-holonomic constraints for parameter estimation of distributed systems. *International Journal of Intelligent Systems Technology and Applications*, 3(3/4):277–295, 2007.

[163] T. Spiess, J. Bange, M. Buschmann, and P. Vörsmann. First application of the meteorological Mini-UAV 'M2AV'. *Meteorologische Zeitschrift*, 16(2):159–169, 2007.

[164] S. Srinivasan. Contour estimation using collaborating mobile sensors. In *Proceedings of the 2006 Workshop on Dependability Issues in Wireless Ad-Hoc Networks and Sensor Networks*, pages 73–82. ACM, 2006.

[165] S. S. Stanković, M. S. Stanković, and D. M. Stipanović. Consensus based overlapping decentralized estimation with missing observations and communication faults. *Automatica*, 45(6):1397–1406, 2009.

[166] G. Steinhauser, A. Brandl, and T. E. Johnson. Comparison of the Chernobyl and Fukushima nuclear accidents: A review of the environmental impacts. *Science of the Total Environment*, 470:800–817, 2014.

[167] J. M. Stockie. The mathematics of atmospheric dispersion modeling. *SIAM Review*, 53(2):349–372, 2011.

[168] A. Stohl, C. Forster, A. Frank, P. Seibert, and G. Wotawa. Technical note: The Lagrangian particle dispersion model FLEXPART version 6.2. *Atmospheric Chemistry and Physics*, 5(9):2461–2474, 2005.

[169] A. Stohl, P. Seibert, G. Wotawa, D. Arnold, J. Burkhart, S. Eckhardt, C. Tapia, A. Vargas, and T. Yasunari. Xenon-133 and caesium-137 releases into the atmosphere from the Fukushima Daiichi nuclear power plant: determination of the source term, atmospheric dispersion, and deposition. *Atmospheric Chemistry and Physics*, 12(5):2313–2343, 2012.

[170] R. Stranders, A. Farinelli, A. Rogers, and N. R. Jennings. Decentralised coordination of mobile sensors using the max-sum algorithm. In *Proceedings of the 21st International Joint Conference on Artifical intelligence*, IJCAI'09, pages 299–304. Morgan Kaufmann Publishers Inc., 2009.

[171] R. B. Stull. *An introduction to boundary layer meteorology*, volume 13. Springer Science & Business Media, 2012.

[172] S. Subchan, B. White, A. Tsourdos, M. Shanmugavel, and R. Zbikowski. Dubins path planning of multiple UAVs for tracking contaminant cloud. In *Proceedings of the 17th World Congress of the International Federation of Automatic Control*, pages 5718–5723, 2008.

[173] S. Susca, F. Bullo, and S. Martinez. Monitoring environmental boundaries with a robotic sensor network. *IEEE Transactions on Control Systems Technology*, 16(2):288–296, 2008.

[174] O. G. Sutton. A theory of eddy diffusion in the atmosphere. *Proceedings of the Royal Society of London. Series A*, 135(826):143–165, 1932.

[175] G. I. Taylor. Diffusion by continuous movements. *Proceedings of the London Mathematical Society*, s2-20(1):196–212, 1922.

[176] TEPCO. Unit 3 of Fukushima Daiichi Nuclear Power Station. `http://photo.tepco.co.jp/en/date/2011/201103-e/110321-01e.html`. Accessed: 2017-01-08.

[177] D. Thomson. Criteria for the selection of stochastic models of particle trajectories in turbulent flows. *Journal of Fluid Mechanics*, 180:529–556, 1987.

[178] M. K. Tippett, J. L. Anderson, C. H. Bishop, T. M. Hamill, and J. S. Whitaker. Ensemble square root filters. *Monthly Weather Review*, 131(7):1485–1490, 2003.

[179] F. Torrisi and M. Baotic. CPLEXINT - Matlab interface for the CPLEX solver.

[180] A. Toselli and O. Widlund. *Domain decomposition methods: algorithms and theory*, volume 34. Springer, 2005.

[181] C. Tricaud and Y.-Q. Chen. Optimal mobile actuator/sensor network motion strategy for parameter estimation in a class of cyber physical systems. In *Proceedings of the American Control Conference (ACC)*, pages 367–372, 2009.

[182] C. Tricaud, M. P. Dariusz, U. Yang, and Q. Chen. D-optimal trajectory design of heterogeneous mobile sensors for parameter estimation of distributed systems. In *Proceedings of the American Control Conference (ACC)*, pages 663–668. IEEE, 2008.

[183] M. Trincavelli, M. Reggente, S. Coradeschi, A. Loutfi, H. Ishida, and A. J. Lilienthal. Towards environmental monitoring with mobile robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2210–2215, 2008.

[184] C. Trudinger, M. Raupach, P. Rayner, and I. Enting. Using the Kalman filter for parameter estimation in biogeochemical models. *Environmetrics*, 19(8):849–870, 2008.

[185] D. B. Turner. *Workbook of atmospheric dispersion estimates: an introduction to dispersion modeling*. CRC press, 1994.

[186] UAS Vision. FMS selects ING Robotic Aviation's fixed wing unmanned aerial vehicle system for forest fire fighting in Chile. `www.uasvision.com/2013/06/17/fms-selects-ing-robotic-aviations-fixed-wing-unmanned-aerial-vehicle-system-for-forest-fire-fighting-in-chile/`. Accessed: 2017-01-08.

[187] D. Uciński. *Measurement optimization for parameter estimation in distributed systems*. Technical University Press, 1999.

[188] D. Ucinski. *Optimal measurement methods for distributed parameter system identification*. CRC Press, 2004.

[189] A. Van den Kroonenberg, S. Martin, F. Beyrich, and J. Bange. Spatially-averaged temperature structure parameter over a heterogeneous surface measured by an unmanned aerial vehicle. *Boundary-Layer Meteorology*, 142(1):55–77, 2012.

[190] R. Viswanathan and P. Varshney. Distributed detection with multiple sensors I. Fundamentals. *Proceedings of the IEEE*, 85(1):54–63, 1997.

[191] E. Walter and L. Pronzato. *Identification of Parametric Models: from Experimental Data*. Springer, 1997 edition, 1997.

[192] G. Welch and G. Bishop. An introduction to the Kalman filter. Technical report, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, 1995.

[193] J. S. Whitaker and T. M. Hamill. Ensemble data assimilation without perturbed observations. *Monthly Weather Review*, 130(7):1913–1924, 2002.

[194] B. A. White, A. Tsourdos, I. Ashokaraj, S. Subchan, and R. Żbikowski. Contaminant cloud boundary monitoring using network of UAV sensors. *Sensors Journal, IEEE*, 8(10):1681–1692, 2008.

[195] Wikipedia. List of atmospheric dispersion models. `https://en.wikipedia.org/wiki/List_of_atmospheric_dispersion_models`. Accessed: 2017-01-08.

[196] J. D. Wilson and B. L. Sawford. Review of Lagrangian stochastic models for trajectories in the turbulent atmosphere. *Boundary-Layer Meteorology*, 78(1-2):191–210, 1996.

[197] J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, 2008.

[198] N. K. Yilmaz, C. Evangelinos, P. F. Lermusiaux, and N. M. Patrikalakis. Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming. *IEEE Journal of Oceanic Engineering*, 33(4):522–537, 2008.

[199] D. Zhang, C. Colburn, and T. Bewley. Estimation and adaptive observation of environmental plumes. In *American Control Conference (ACC)*, pages 4281–4286, 2011.

[200] F. Zhang and N. E. Leonard. Generating contour plots using multiple sensor platforms. In *SIS*, pages 309–316, 2005.

[201] H. Zhou, J. J. Gomez-Hernandez, H.-J. H. Franssen, and L. Li. An approach to handling non-Gaussianity of parameters and state variables in ensemble Kalman filtering. *Advances in Water Resources*, 34(7):844–864, 2011.

[202] T. Zięba and D. Uciński. Mobile sensor routing for parameter estimation of distributed systems using the parallel tunneling method. *International Journal of Applied Mathematics and Computer Science*, 18(3):307–318, 2008.

Own Publications

Peer-Reviewed Papers

Tobias Ritter, Stefan Ulbrich, and Oskar von Stryk. Decentralized dynamic data-driven monitoring of dispersion processes on decomposed domains. *Procedia Computer Science*, 108:1632–1641, 2017.

Tobias Ritter, Juliane Euler, Stefan Ulbrich, and Oskar von Stryk. Decentralized dynamic data-driven monitoring of atmospheric dispersion processes. *Procedia Computer Science*, 80:919–930, 2016.

Juliane Euler, Tobias Ritter, Stefan Ulbrich, and Oskar von Stryk. Centralized ensemble-based trajectory planning of cooperating sensors for estimating atmospheric dispersion processes. In Sai Ravela and Adrian Sandu, editors, *Dynamic Data-Driven Environmental Systems Science*, volume 8964 of *Lecture Notes in Computer Science*, pages 322–333. Springer International Publishing, 2015.

Tobias Ritter, Juliane Euler, Stefan Ulbrich, and Oskar von Stryk. Adaptive observation strategy for dispersion process estimation using cooperating mobile sensors. In *Proceedings of the 19th IFAC World Congress*, pages 5302 – 5308, Cape Town, South Africa, 2014.

Workshop Papers

Tobias Ritter, Stefan Ulbrich, and Oskar von Stryk. Decentralized dynamic data-driven monitoring and estimation of contaminant plumes. In *DGR Days 2016*, page 39, 2016.

Tobias Ritter, Stefan Ulbrich, and Oskar von Stryk. Reduced order models for distributed adaptive monitoring of atmospheric dispersion processes. In *Model Reduction of Parametrized Systems III*, page 84, 2015.

Tobias Ritter, Juliane Euler, Stefan Ulbrich, and Oskar von Stryk. Centralized adaptive observation strategy for atmospheric dispersion process estimation using mobile sensors. In *3rd Workshop on Computational Engineering*, pages 221–224, 2014.

Wissenschaftlicher Werdegang[1]

2007	Allgemeine Hochschulreife
2007 - 2013	Studium Compuational Engineering, Technische Universität Darmstadt
09/2010	B.Sc. in Computational Engineering
08/2013	M.Sc. in Computational Engineering
2011 - 2015	Fast-Track-Stipendiat, Graduate School Computational Engineering
seit 2013	Doktorand, Fachgebiet Simulation, Systemoptimierung und Robotik, Fachbereich Informatik, Technische Universität Darmstadt

Erklärung[2]

Hiermit erkläre ich, dass ich die vorliegende Arbeit, mit Ausnahme der ausdrücklich genannten Hilfsmittel, selbständig verfasst habe.

[1] gemäß § 20 Abs. 3 der Promotionsordnung der TU Darmstadt
[2] gemäß § 9 Abs. 1 der Promotionsordnung der TU Darmstadt